tinyML® Research Symposium

Enabling Ultra-low Power Machine Learning at the Edge

April 22, 2024

www.tinyML.org

When Considering New Hardware Ideas, Build Complete ML Systems!

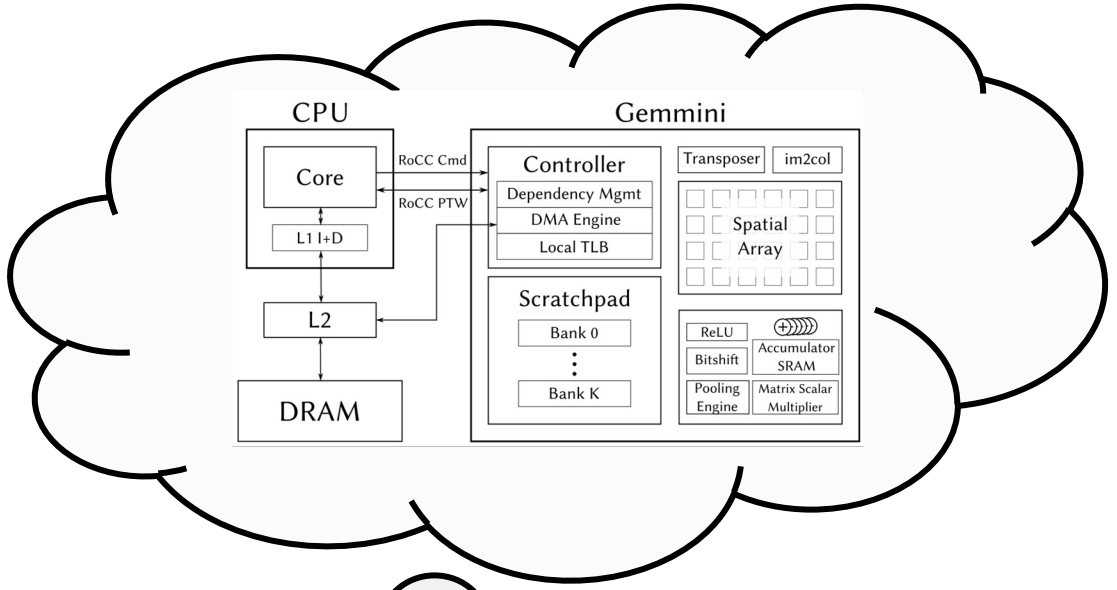
Borivoje Nikolić
bora@berkeley.edu
University of California, Berkeley

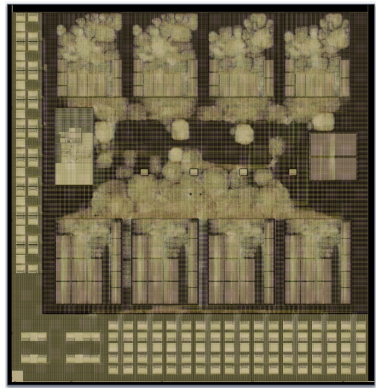
Berkeley
UNIVERSITY OF CALIFORNIA

So, You Have a Brilliant New Idea

TINY

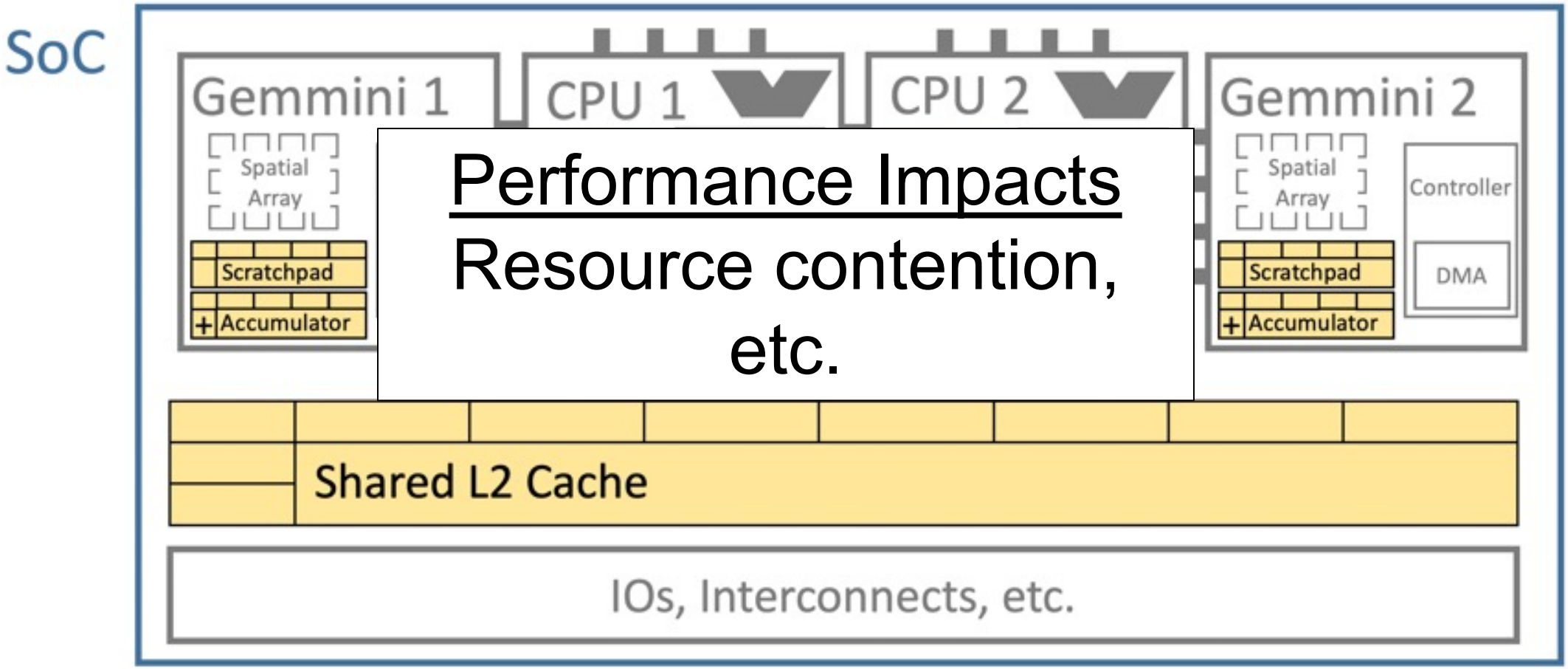
+



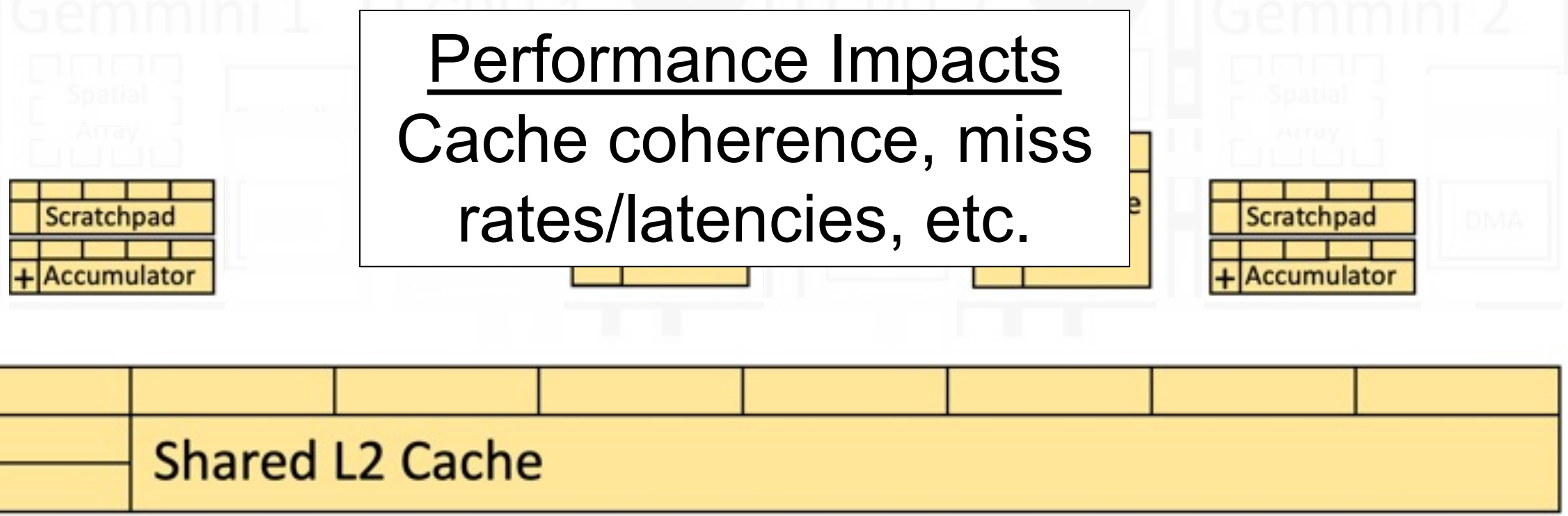


Tiny chip

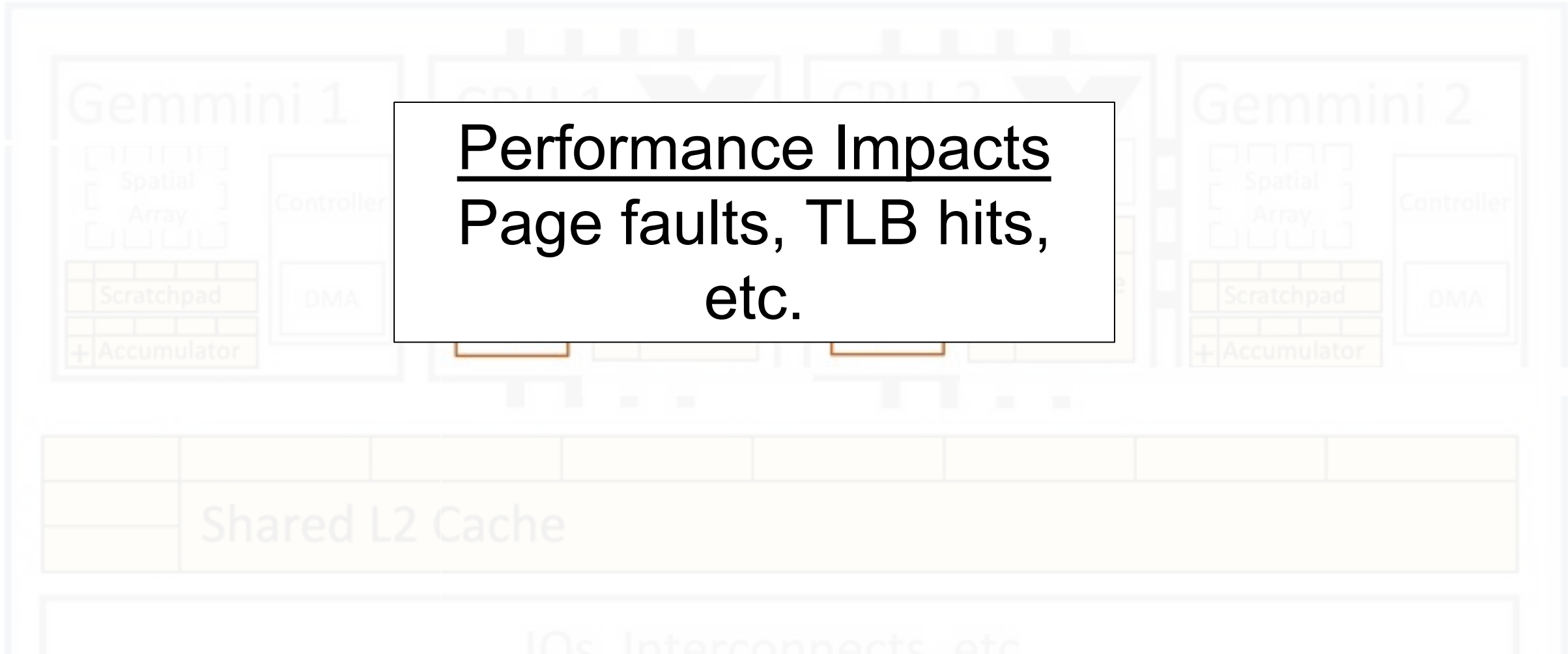
Full-System Insight: SoC



Full-System Visibility: Memory Hierarchy



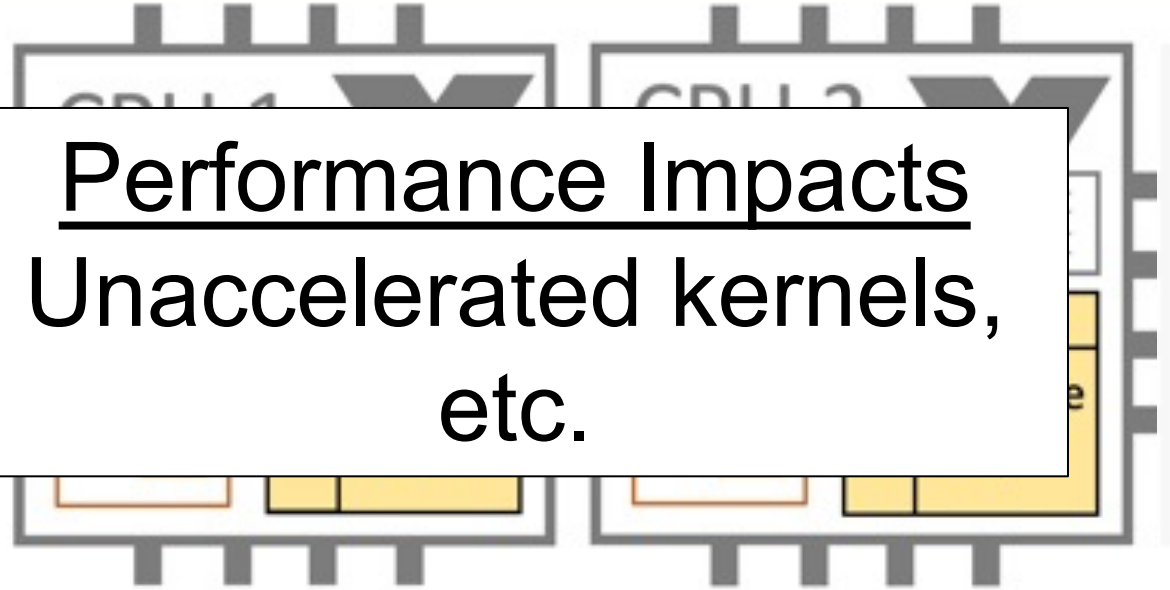
Full-System Visibility: Virtual Addresses



Performance Impacts

Page faults, TLB hits, etc.

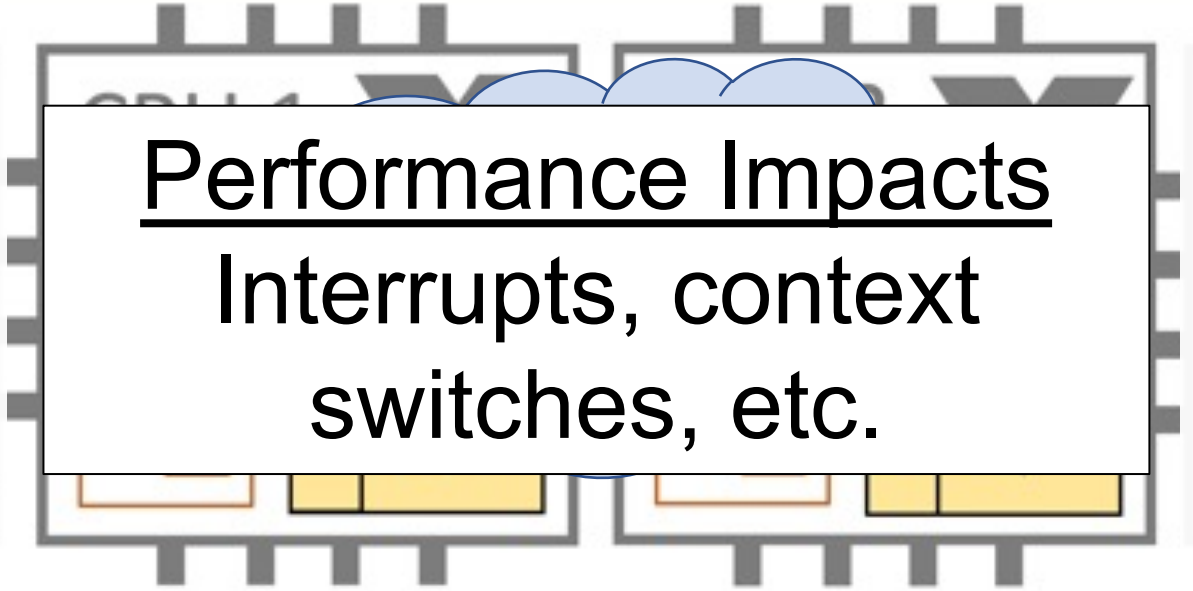
Full-System Visibility: Host CPUs



Performance Impacts

Unaccelerated kernels, etc.

Full-System Visibility: Operating System



Enter CHIPYARD

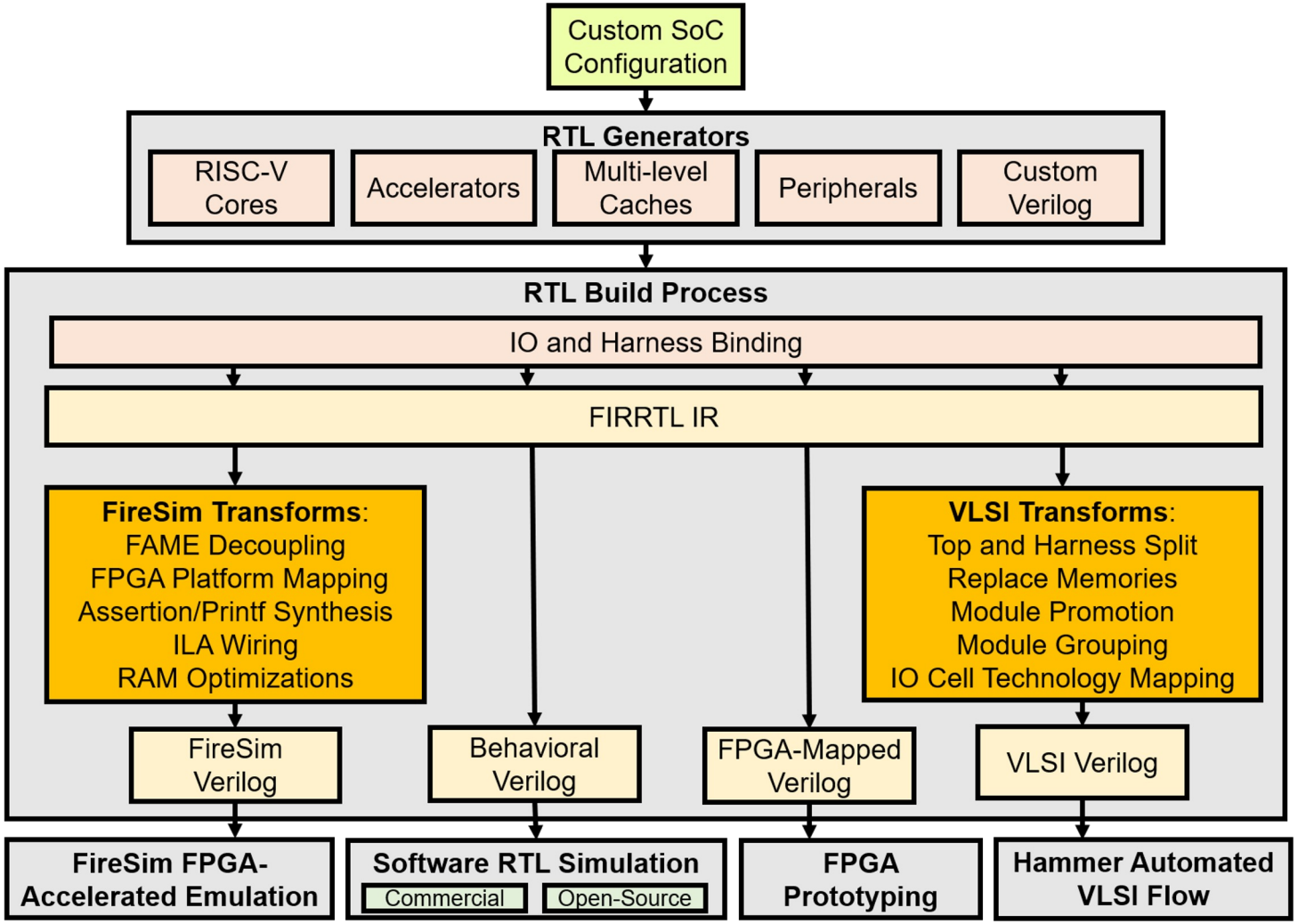
Custom SoC/SiP architecture
New modules + reusing existing modules

- **RTL Simulation** running test binaries/micro-benchmarks
- **FPGA-accelerated simulation** running full workloads
- **FPGA prototyping** for fast demos
- **Tape-out an SoC** prototype

CHIPYARD Organization

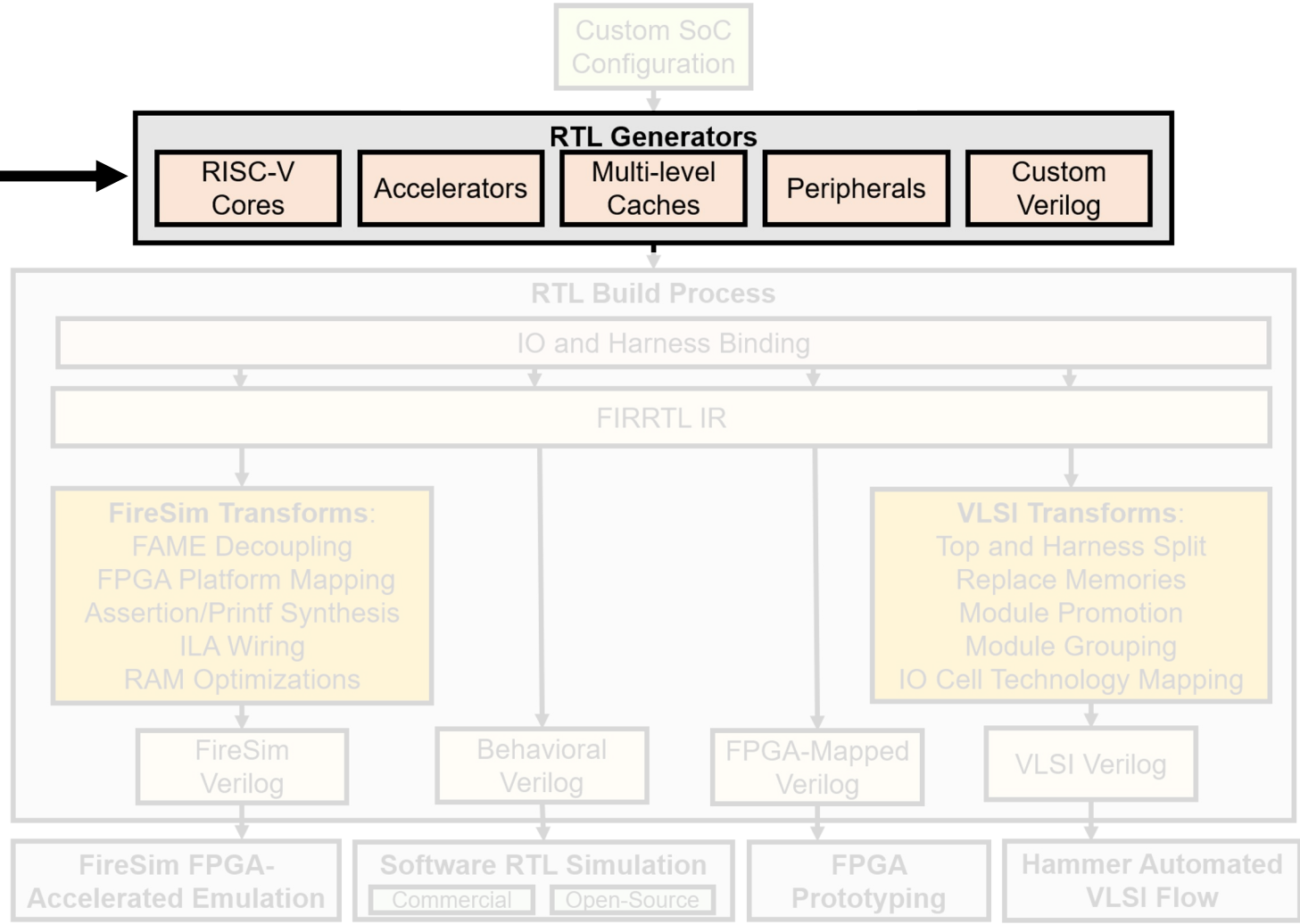
What is Chipyard?

- An organized **framework** for various SoC/SiP design tools and software
- A **curated IP library** of open-source RISC-V SoC components
- A **methodology** for agile SoC/SiP architecture design, exploration, and evaluation

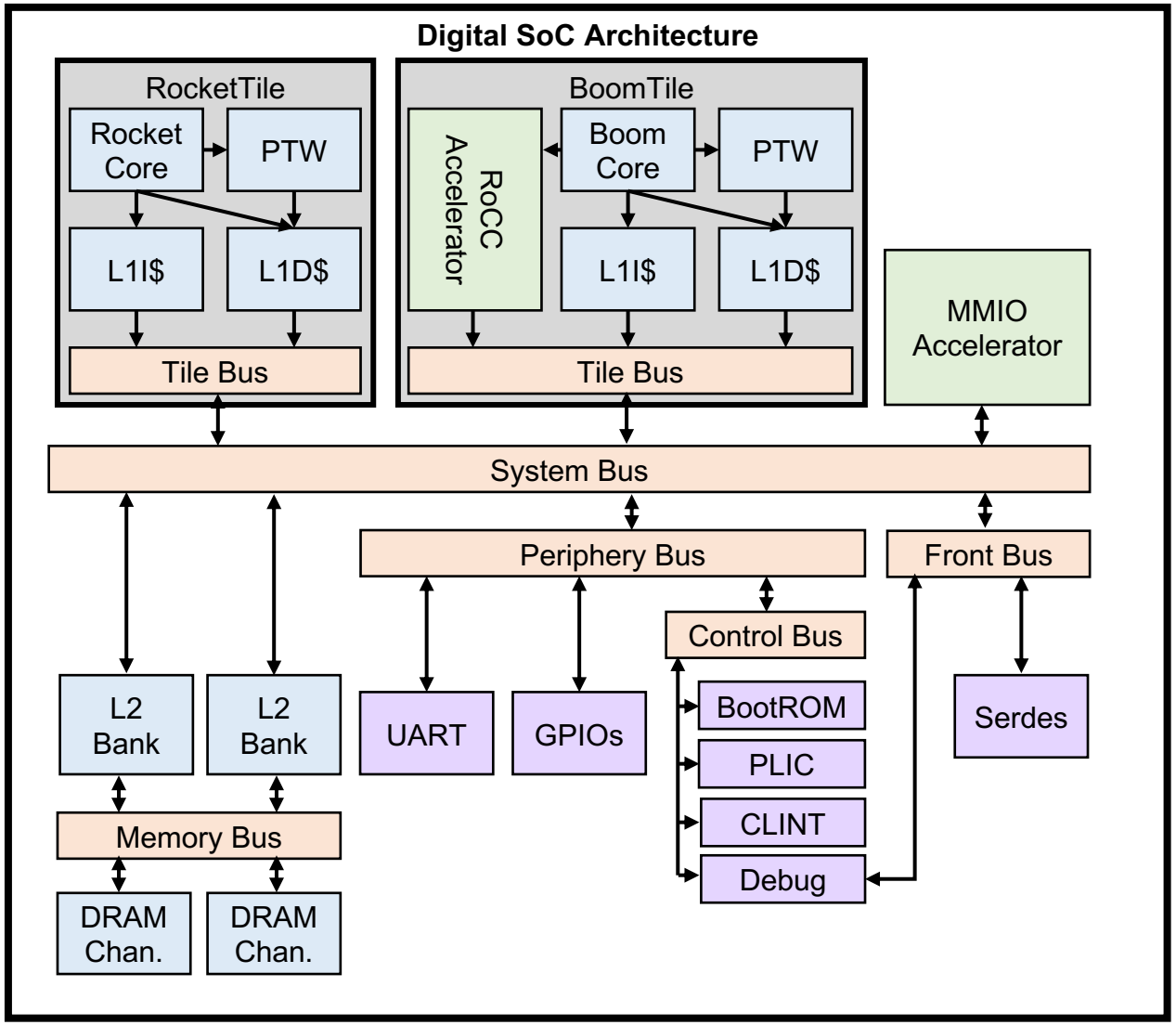


CHIPYARD Organization

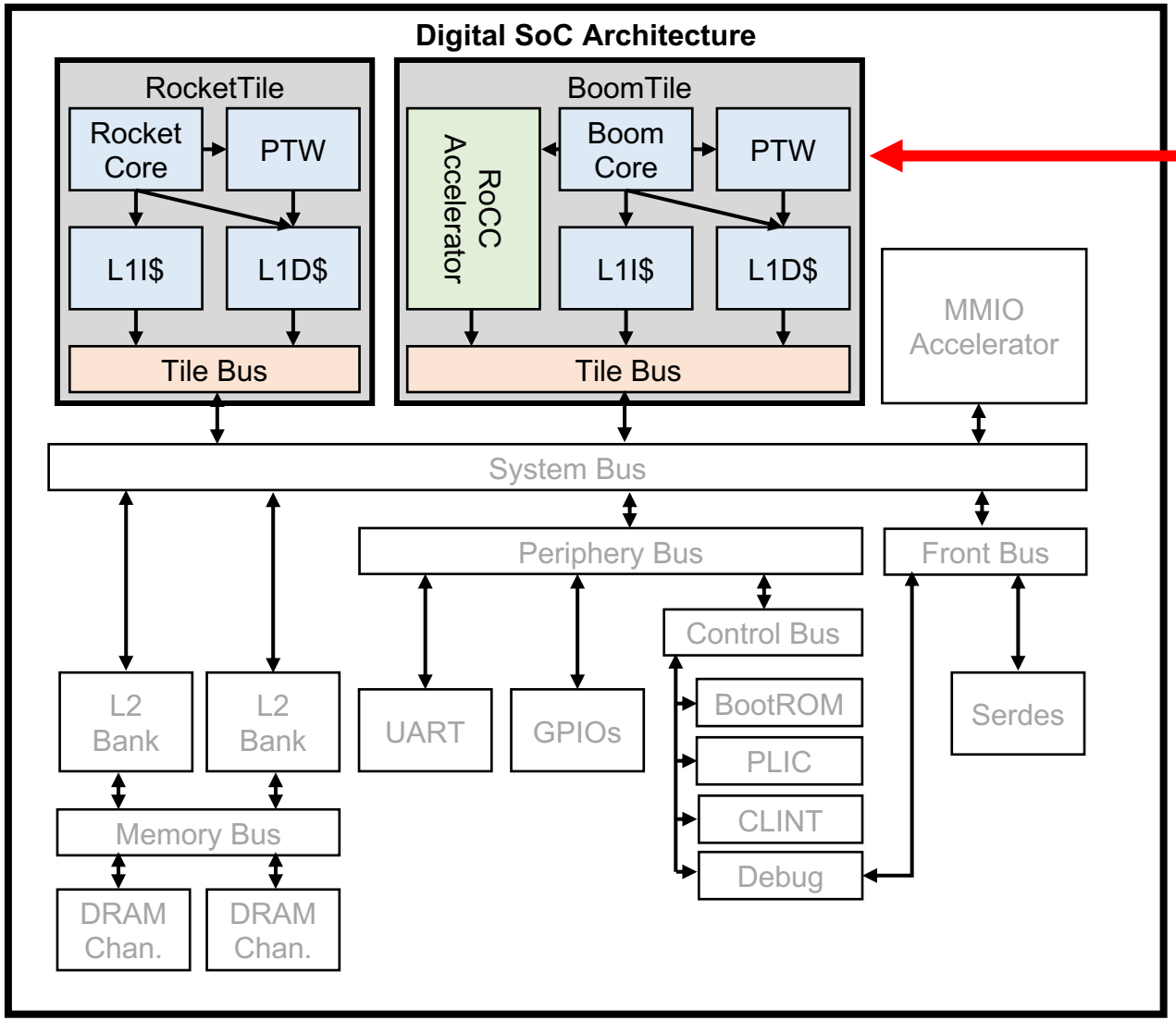
SoC/SiP architecture and generators



SoC Architecture



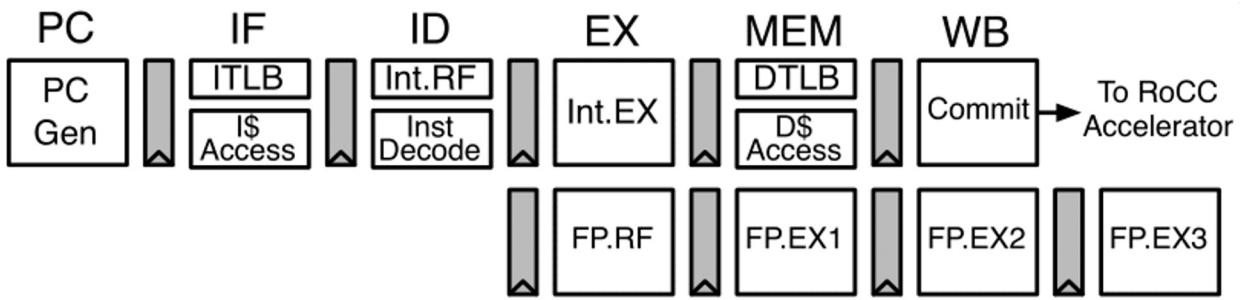
Tiles and Cores



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

Rocket and BOOM

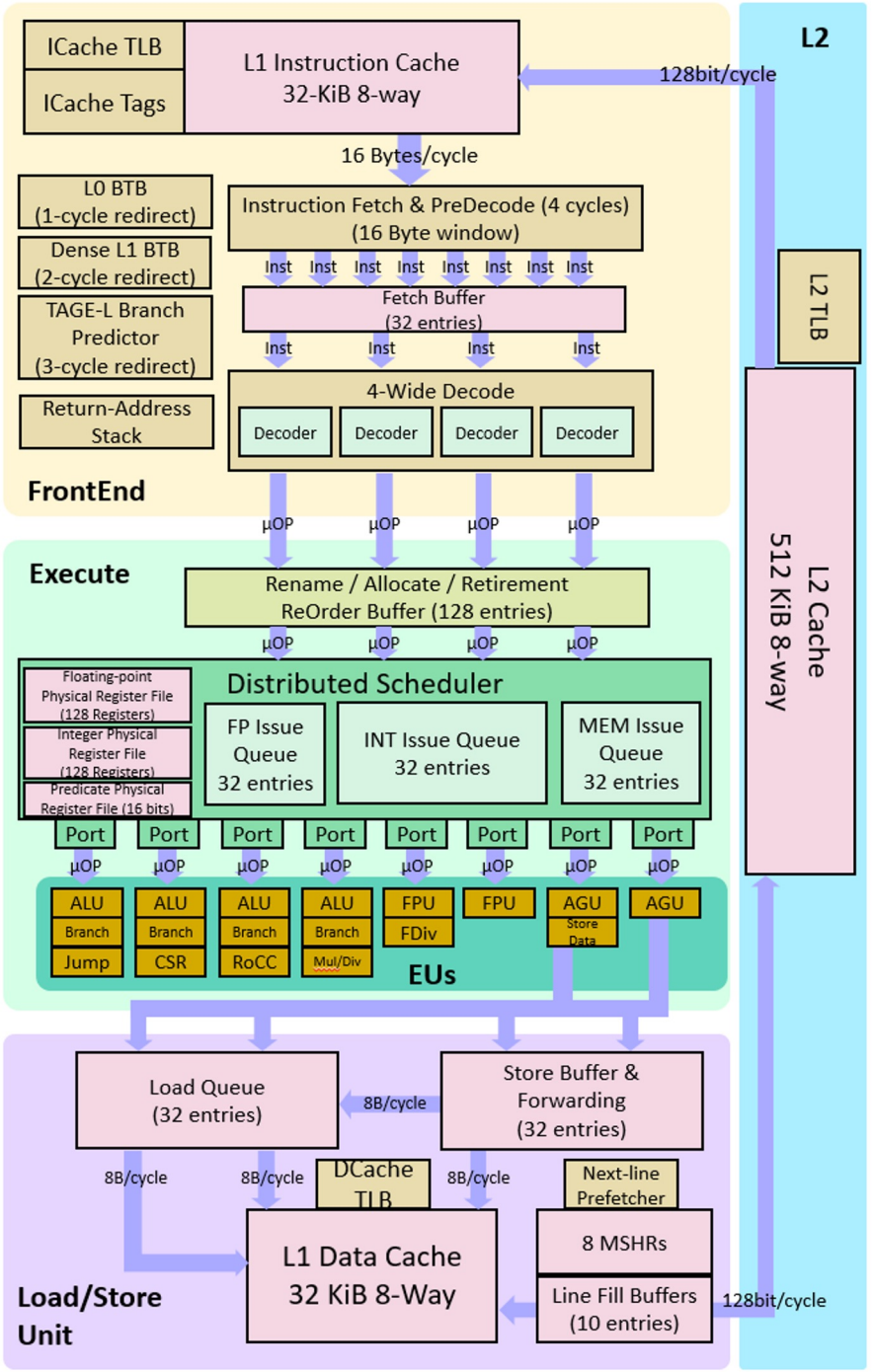


Rocket:

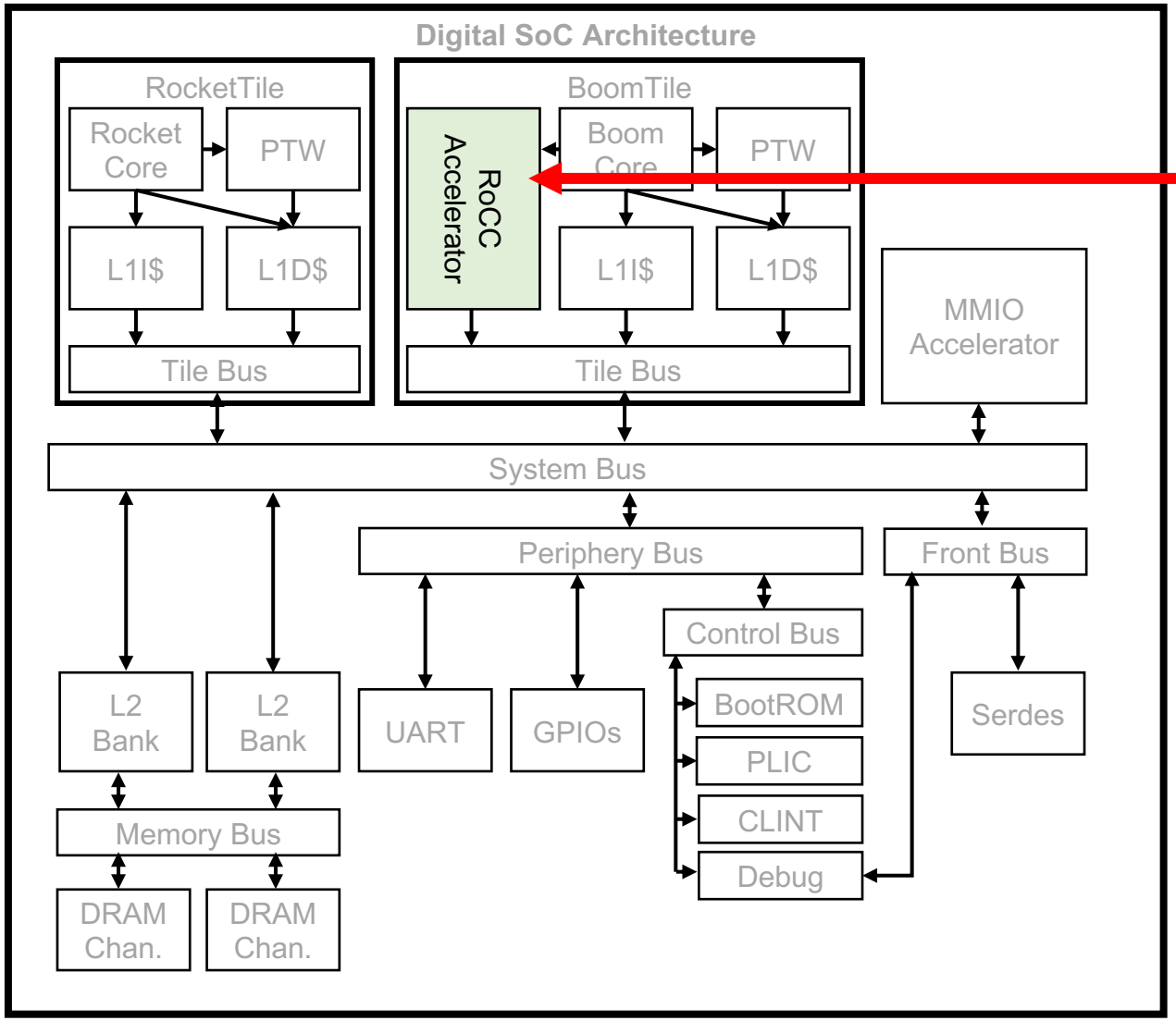
- First open-source RISC-V CPU
- In-order, single-issue RV64GC core
- Efficient design point for low-power devices

SonicBOOM:

- Superscalar out-of-order RISC-V CPU
- Advanced microarchitectural features to maximize IPC
- TAGE branch prediction, OOO load-store-unit, register renaming
- High-performance design for general-purpose systems



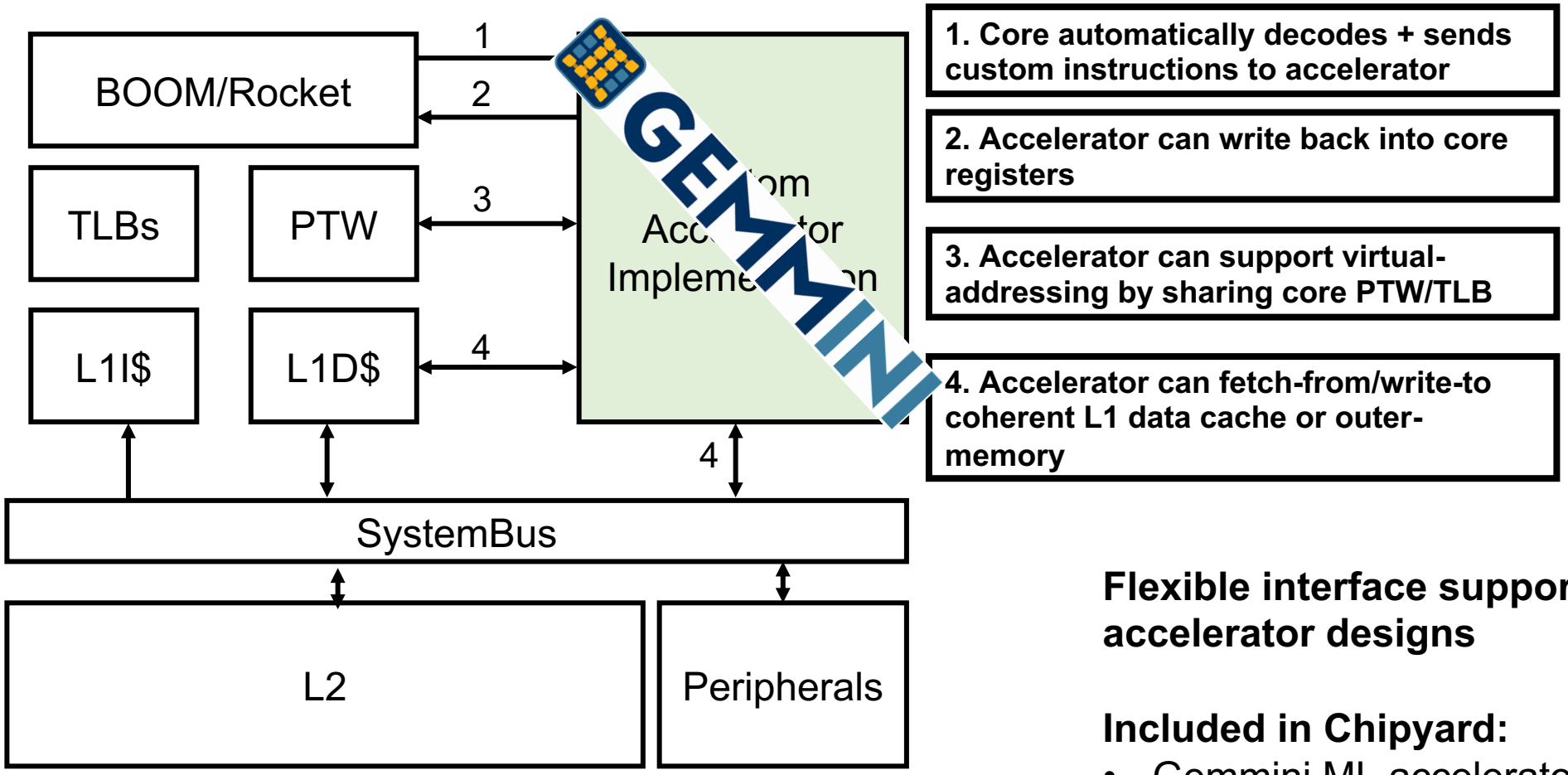
RoCC Accelerators



RoCC Accelerators:

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

RoCC Accelerators

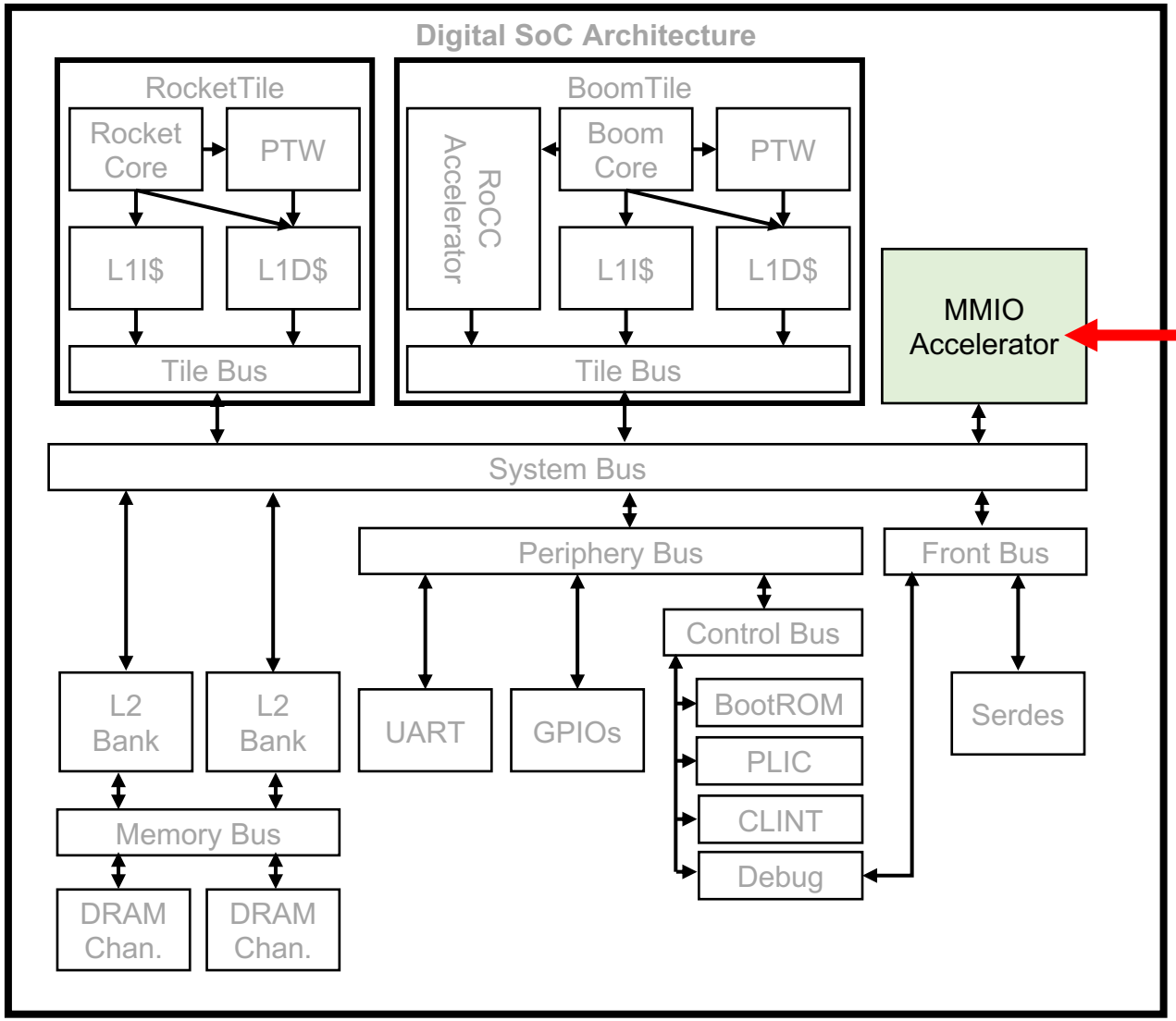


Flexible interface supports a variety of accelerator designs

Included in Chipyard:

- Gemmini ML accelerator
- Hwacha vector accelerator
- SHA3 accelerator

MMIO Accelerators



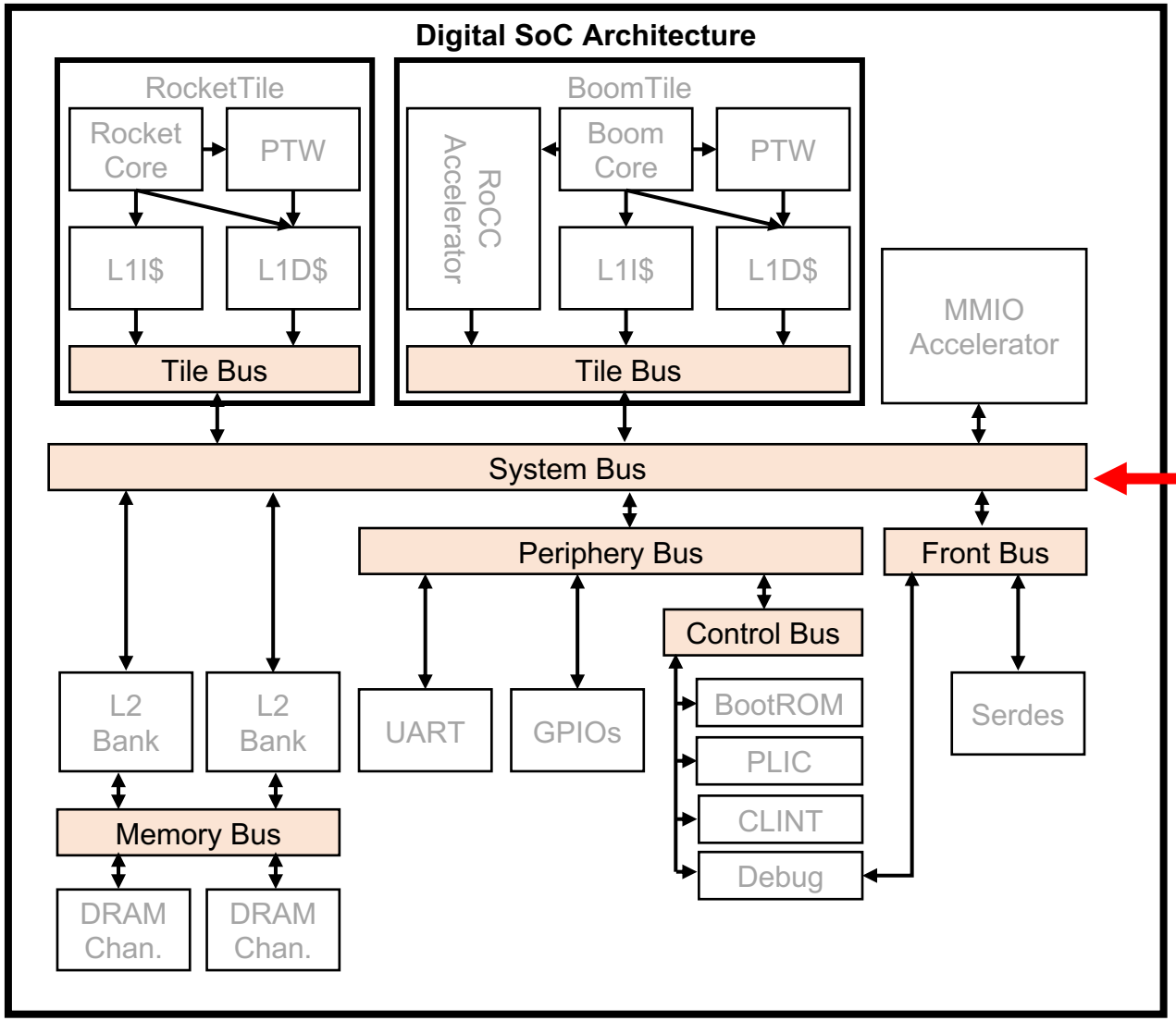
MMIO Accelerators:

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

NVIDIA

NVDLA.org

Coherent Interconnect



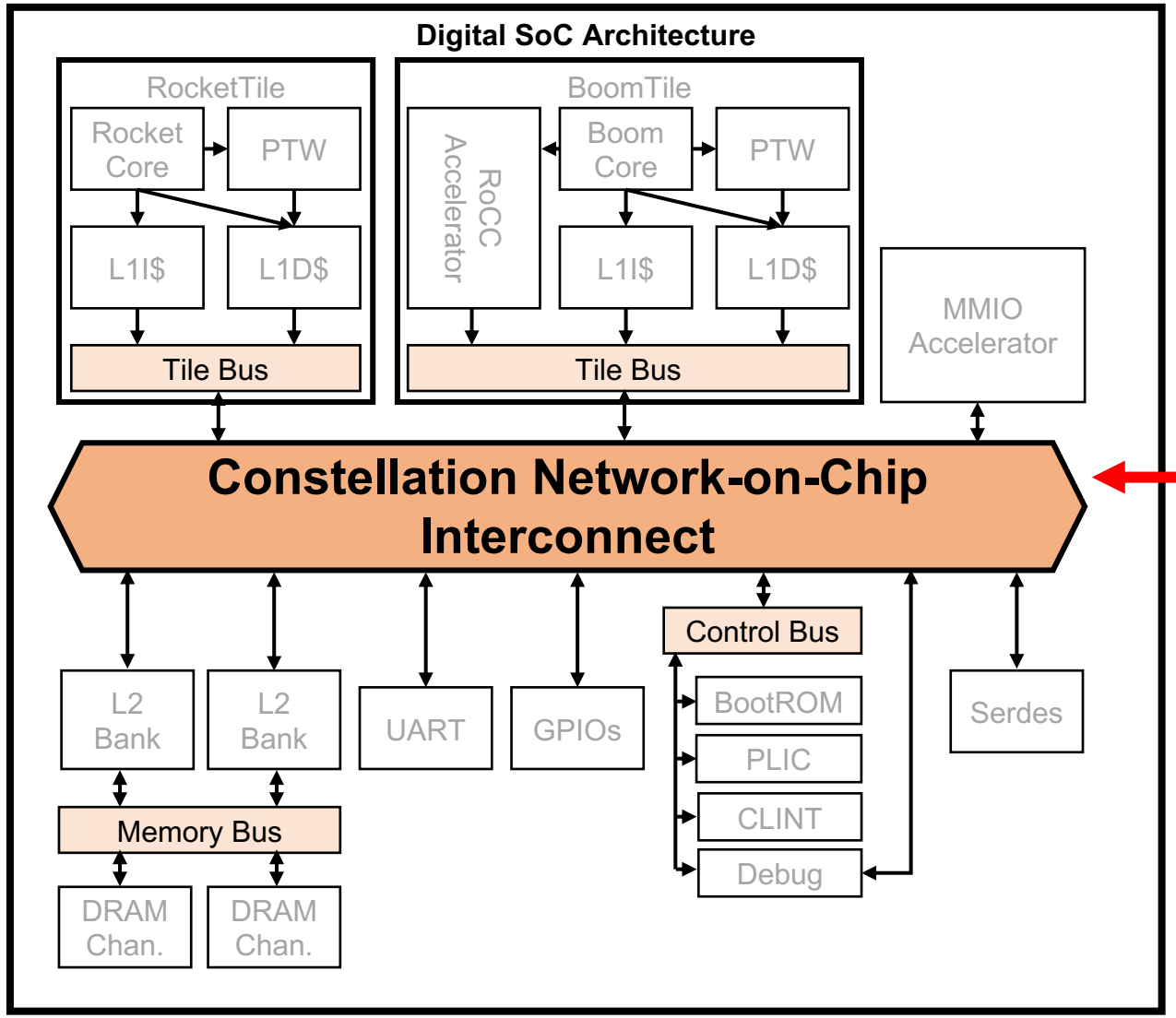
TileLink Standard:

- 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

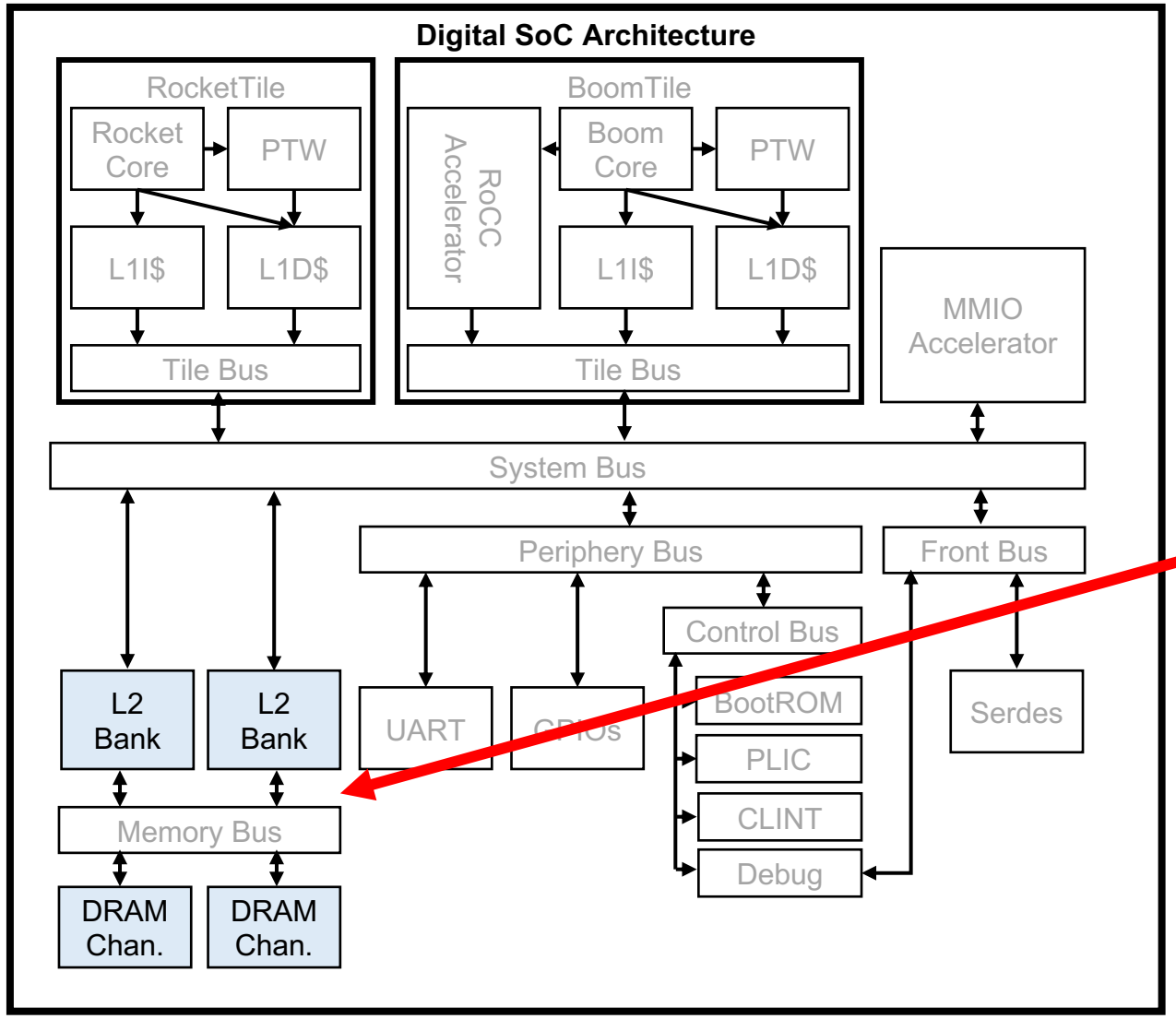
NoC Interconnect



Constellation

- Flexible NoC generator
- Drop-in replacement for TileLink crossbars

L2/DRAM



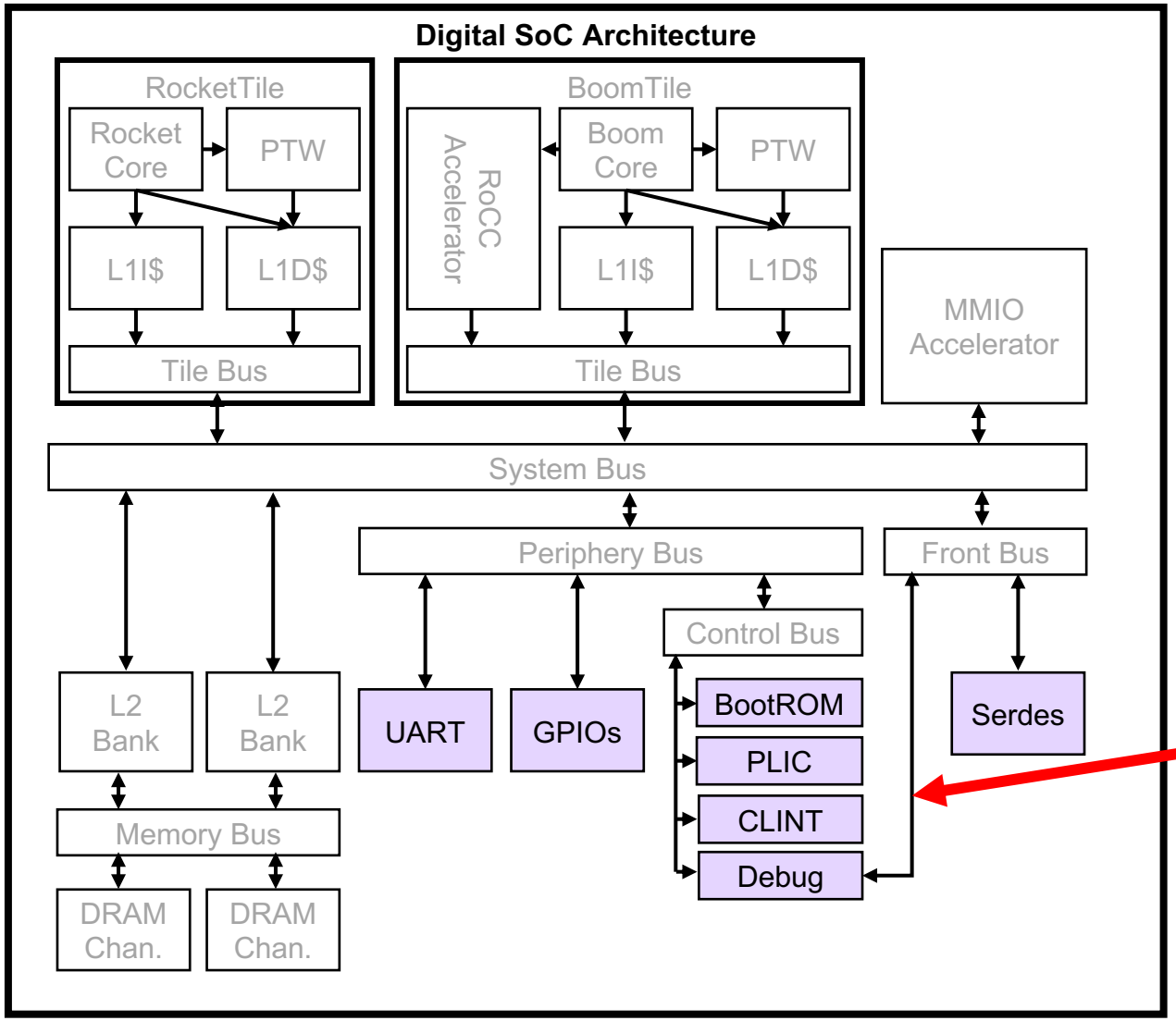
Shared memory:

- Open-source TileLink L2 developed by SiFive
  - Directory-based coherence with MOESI-like protocol
  - Configurable capacity/banking
- Support broadcast-based coherence in no-L2 systems
- Support incoherent memory systems

DRAM:

- AXI-4 DRAM interface to external memory controller
- Interfaces with DRAMSim

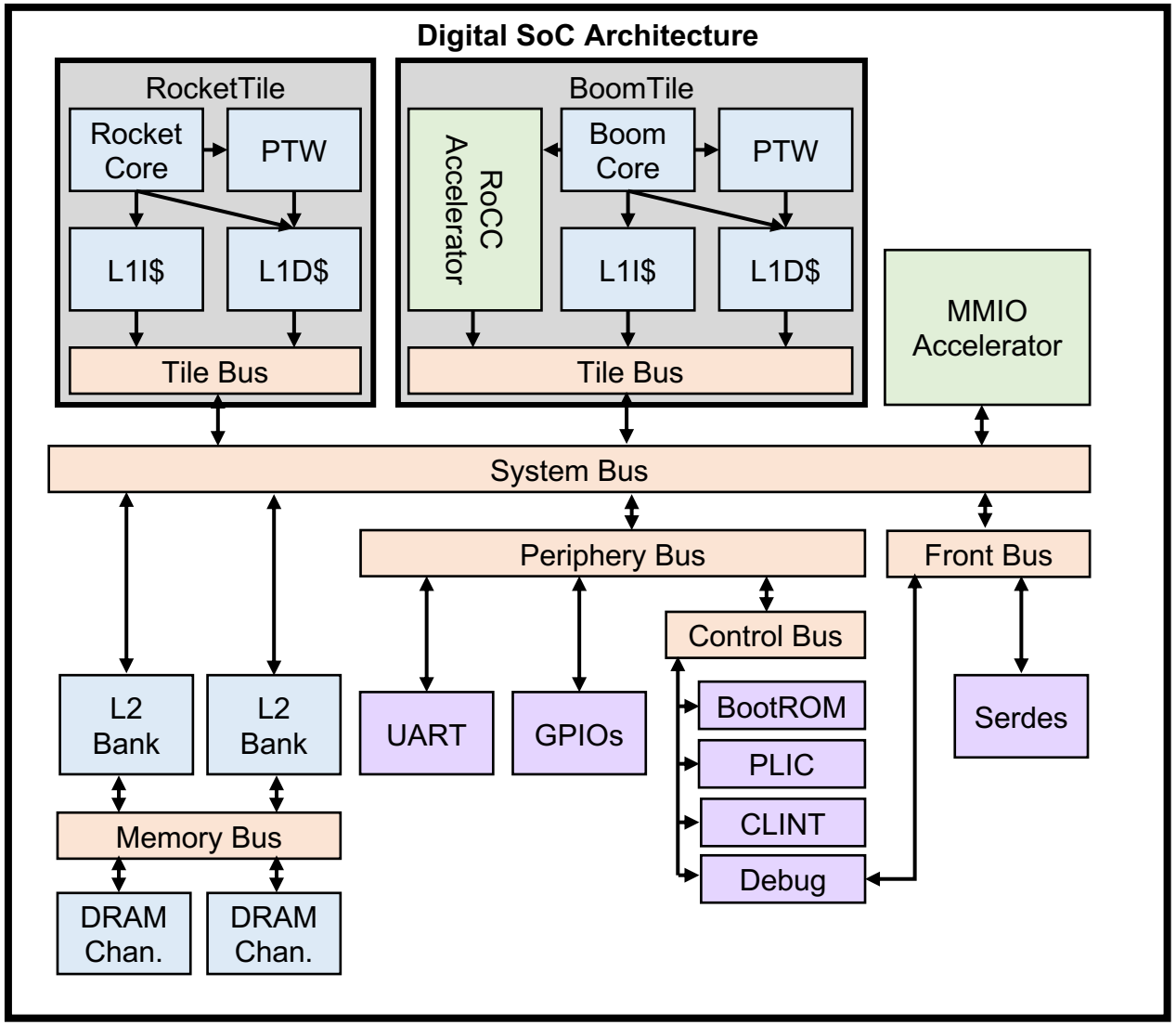
Peripherals and IO



Peripherals and IO:

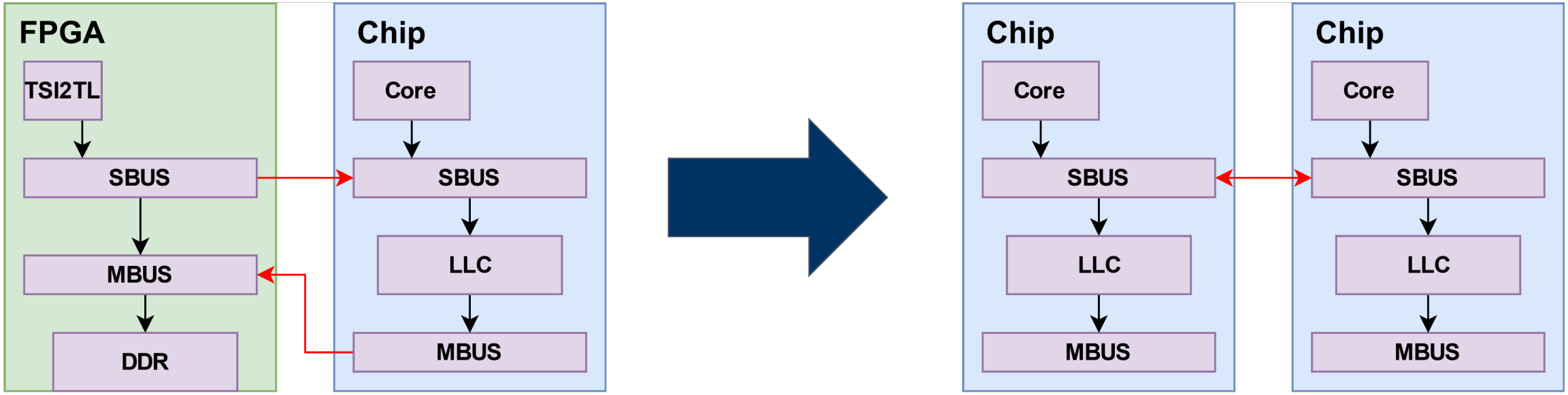
- Open-source RocketChip blocks
  - Interrupt controllers
  - JTAG, Debug module, BootROM
  - UART, GPIOs, SPI, I2C, PWM, etc.
- TestChipIP: useful IP for test chips
  - Clock-management devices
  - SerDes
  - Scratchpads

SoC Architecture



Extending to Chiplets

Take advantage of existing IP
Extend bringup infrastructure
- Configurable bus connections off-chip
- Make chip<->FPGA APIs generic

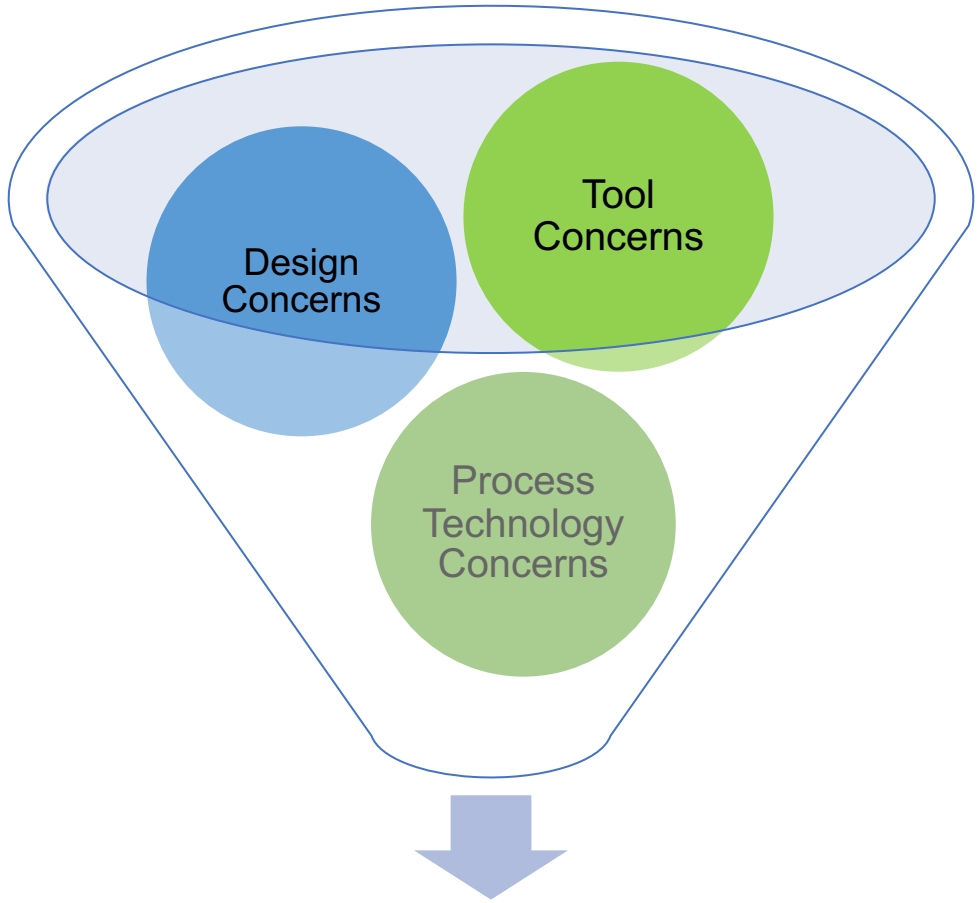


CHIPYARD

CHIPLETYARD

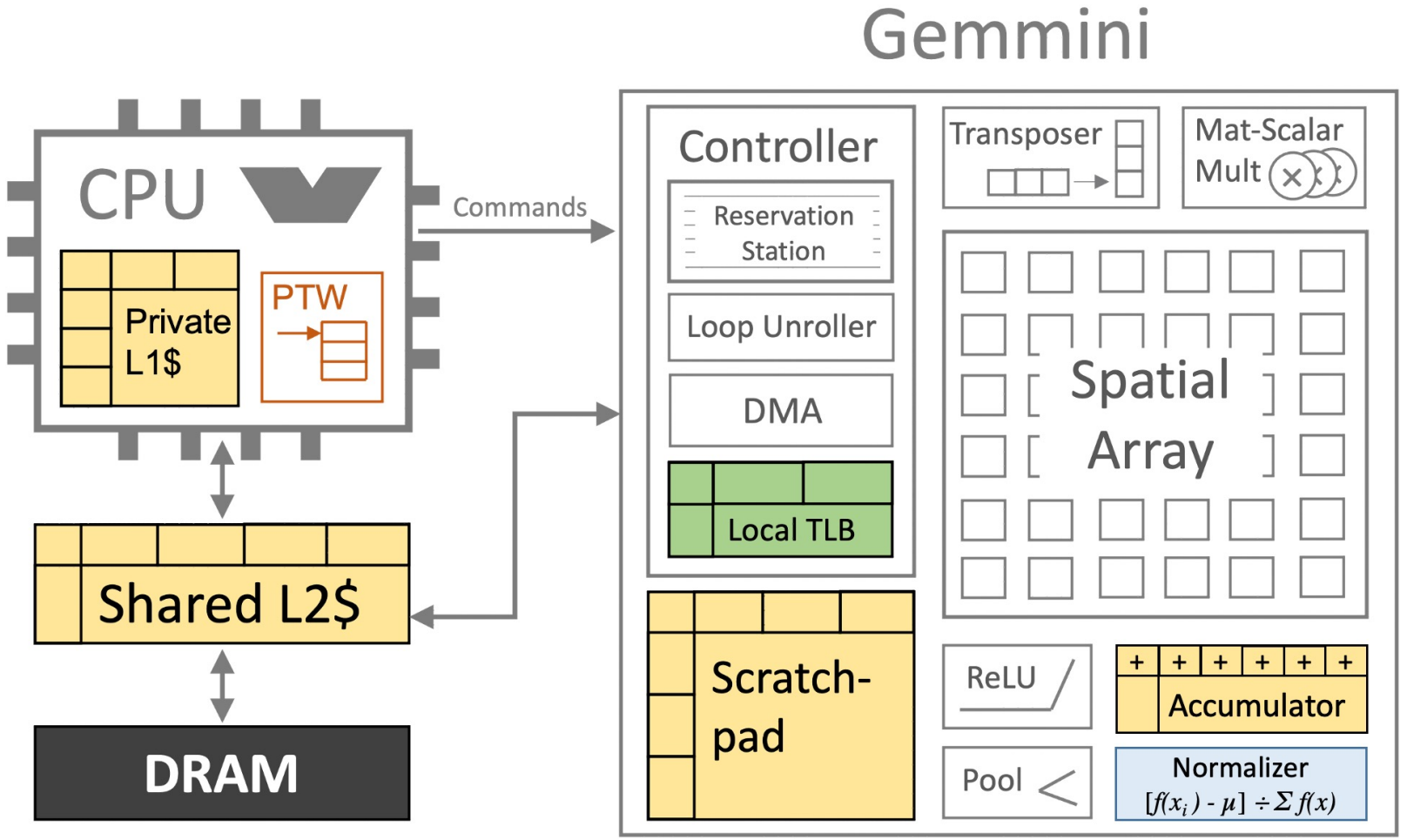
Flow Tool: HAMMER

- 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



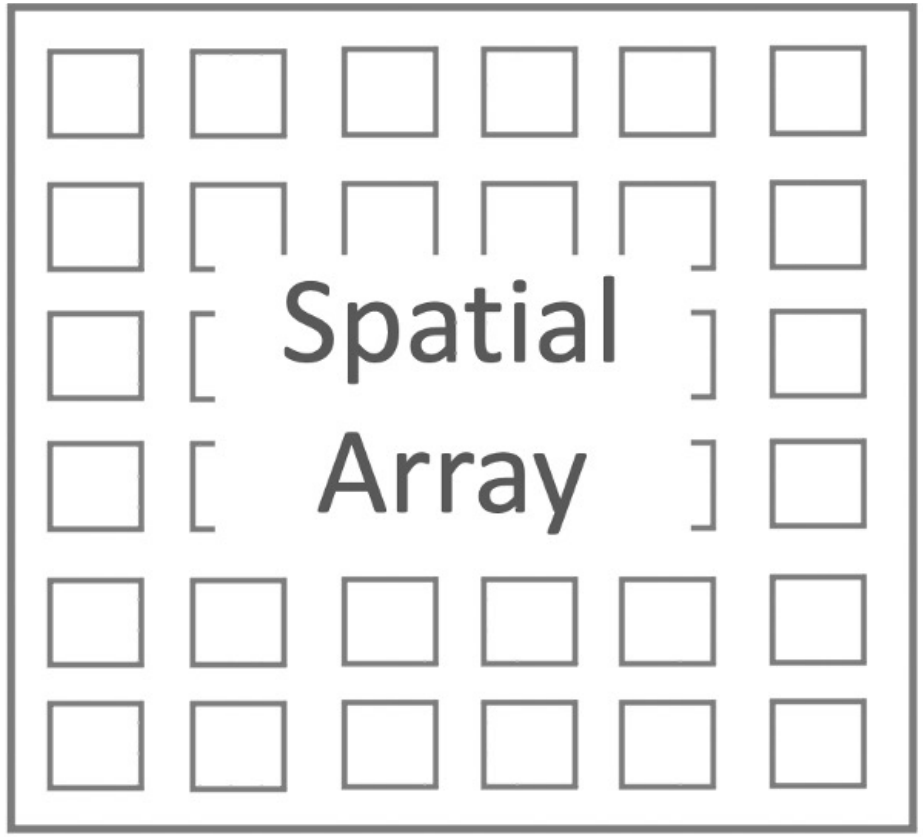
Gemmini in CHIPYARD

- DNN accelerator generator
- Flexible hardware template
- Full-stack
- Full-system



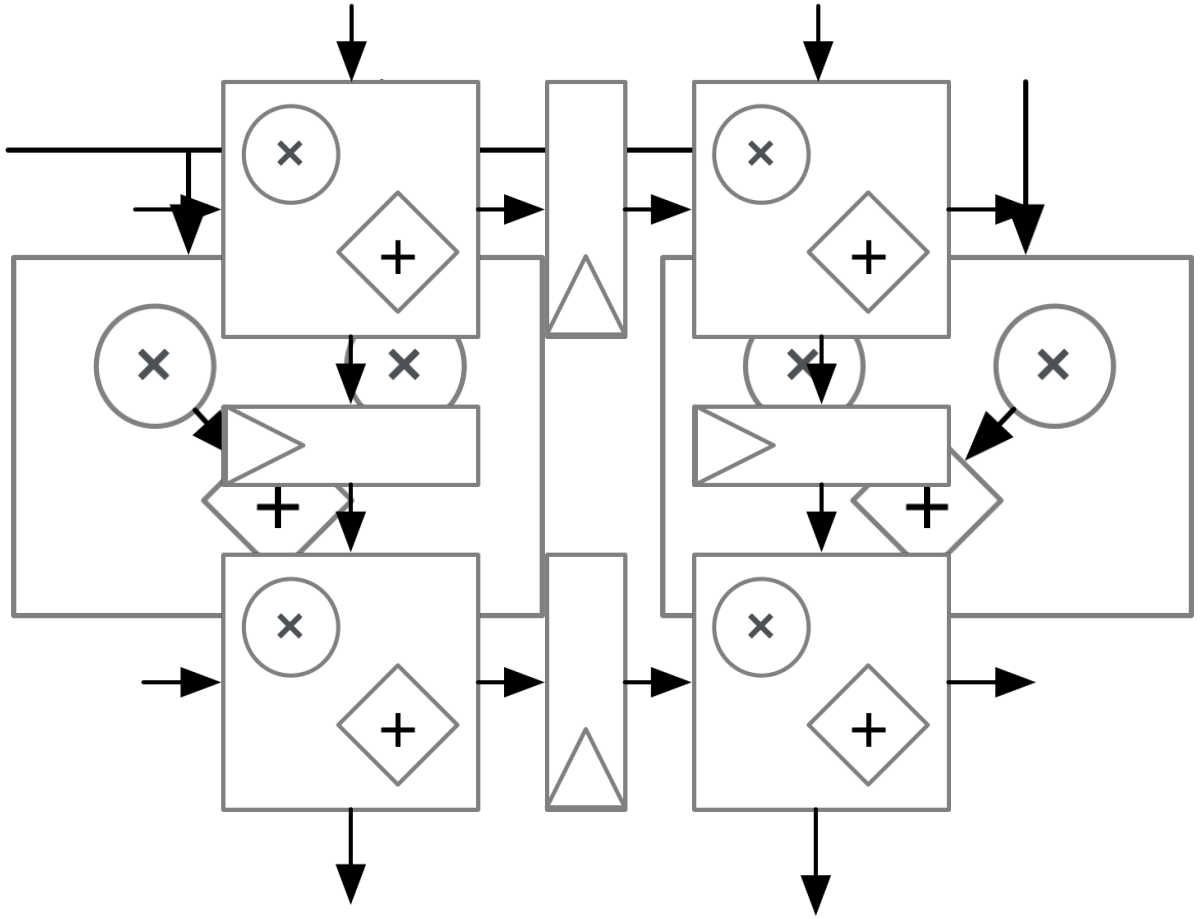
GEMMINI: Spatial Array for Matrix Algebra

- Parameters:
  - Dataflow
  - Dimensions
  - Datatypes
  - Pipelining



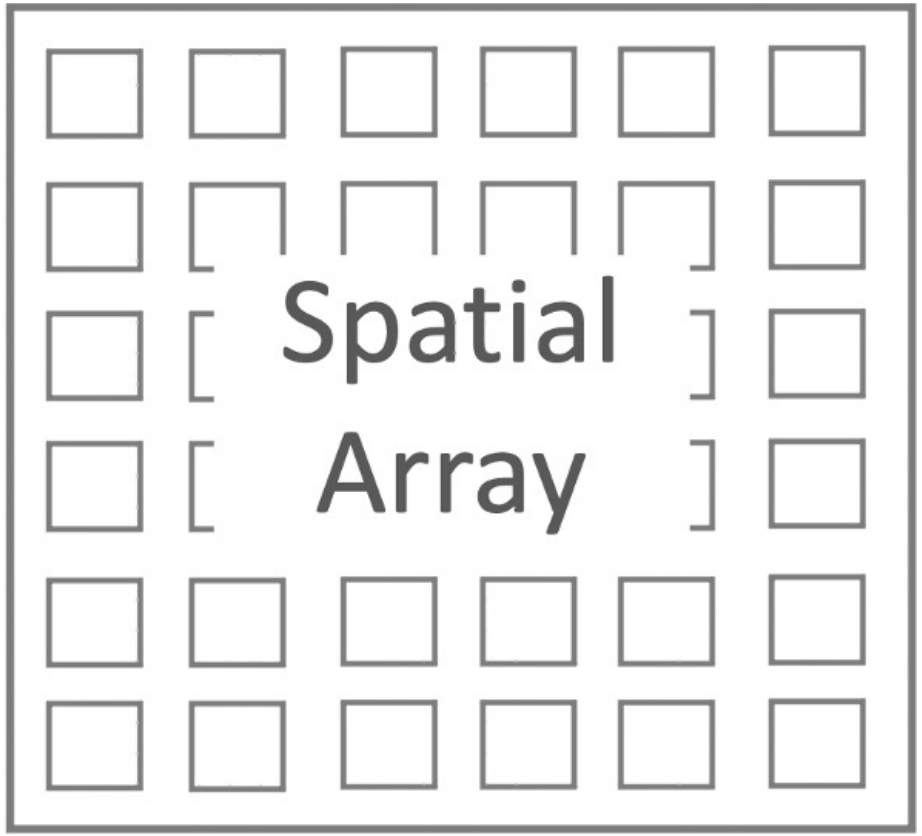
Gemmini: Spatial Array for Matrix Algebra

- Parameters:
  - Dataflow
  - Dimensions
  - Datatypes
  - Pipelining



Gemmini: Spatial Array for Matrix Algebra

- Parameters:
  - Dataflow
  - Dimensions
  - Datatypes
  - Pipelining



Gemmini: Non-GEMM Functionality

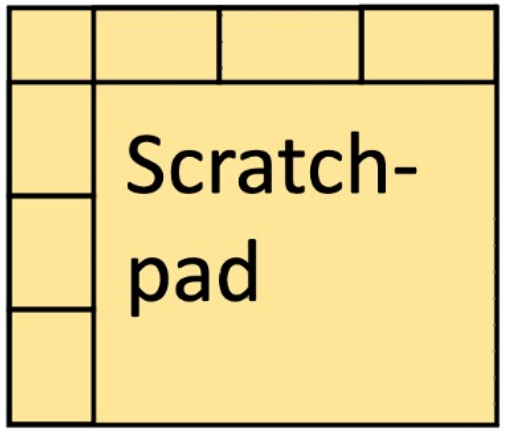
- Can be optimized out at elaboration-time
  - Softmax
  - Layernorm
  - Activation functions
  - Max-pool
  - Transpositions
  - Matrix-scalar operations

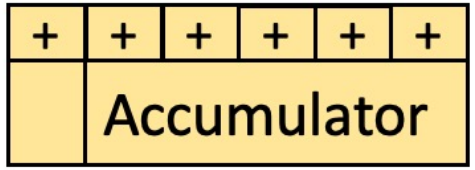
Gemmini: Loop Unroller

- Dynamically schedules operations
  - Such as on-the-fly im2Col
- Parameters:
  - Types of loops to unroll in hardware
    - Only inference kernels?
    - Training kernels too?

Gemmini: Local Scratchpad and Accumulator

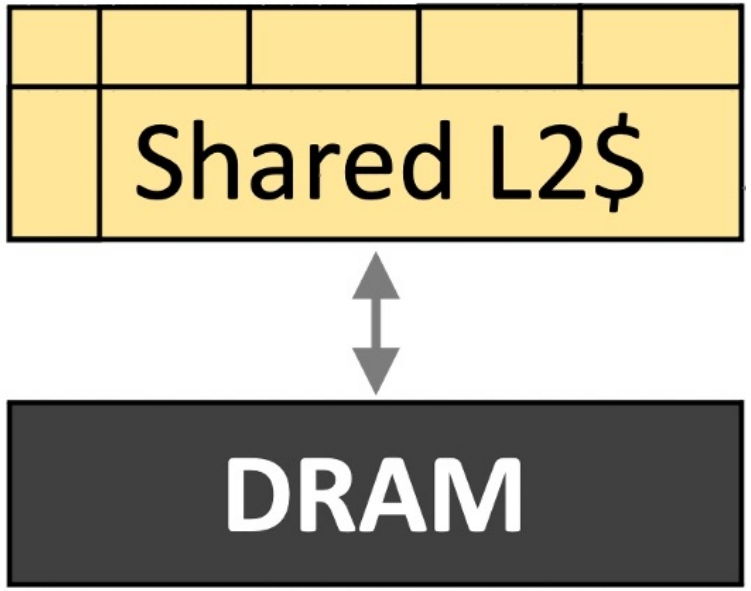
- Parameters:
  - Capacity
  - Banks
  - Single- or dual-port





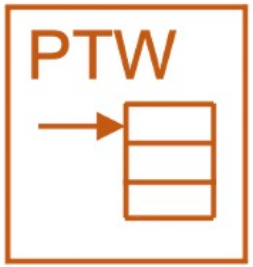
Gemmini: System Memory

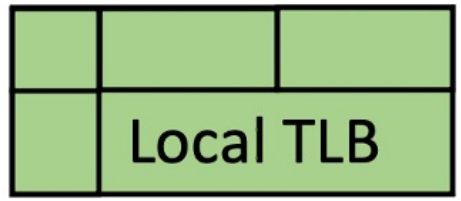
- Parameters:
  - Capacity
  - Banks
  - Optional L3
  - DRAM controller



Gemmini: Virtual Address Translation

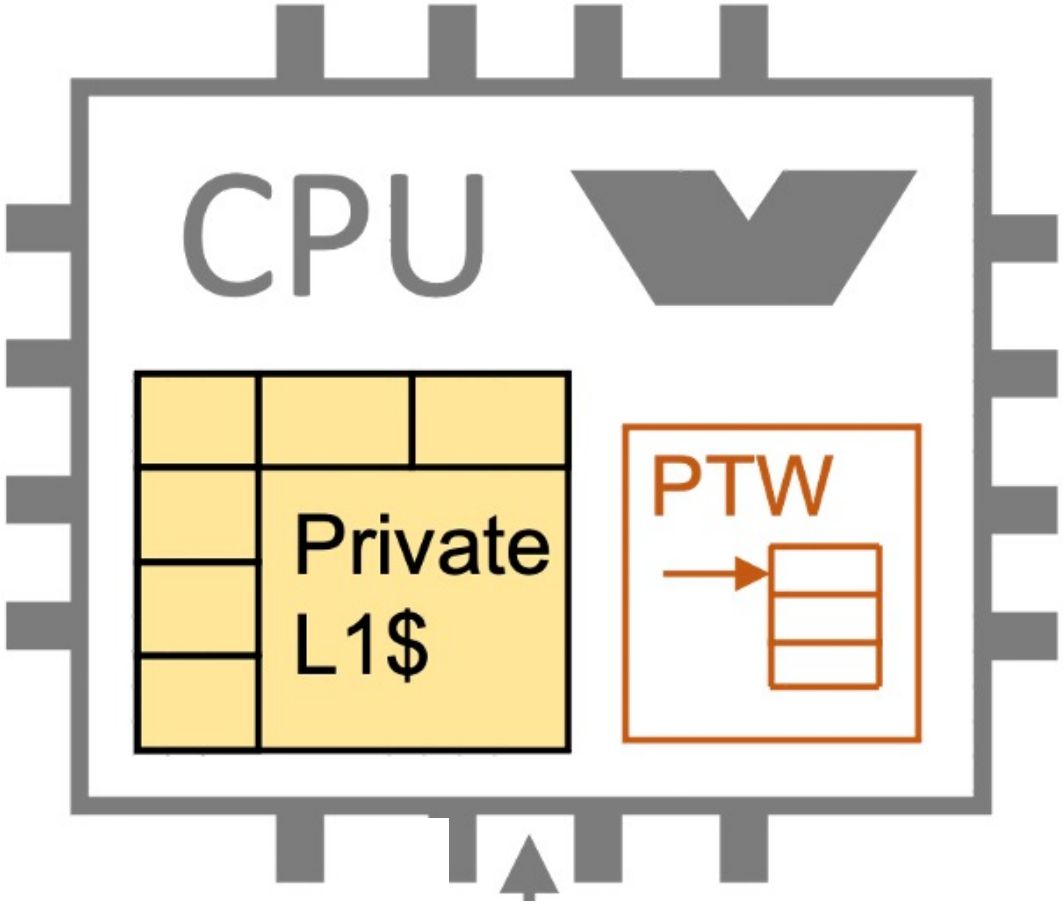
- Parameters:
  - TLB capacity
  - TLB hierarchy
    - e.g. L2 TLB



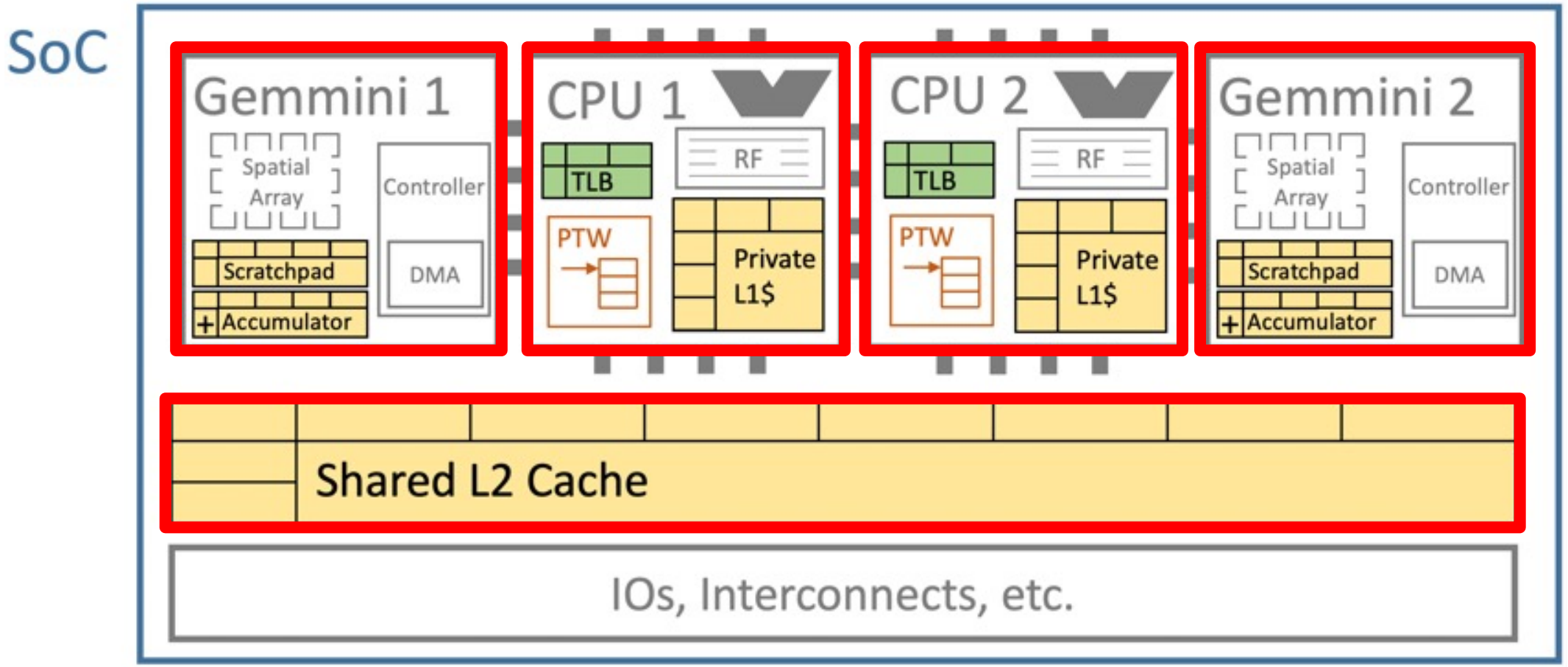


Gemmini: Host CPU

- Parameters:
  - In-order/out-of-order
  - ROB capacity
  - L1 capacity
  - Branch predictor



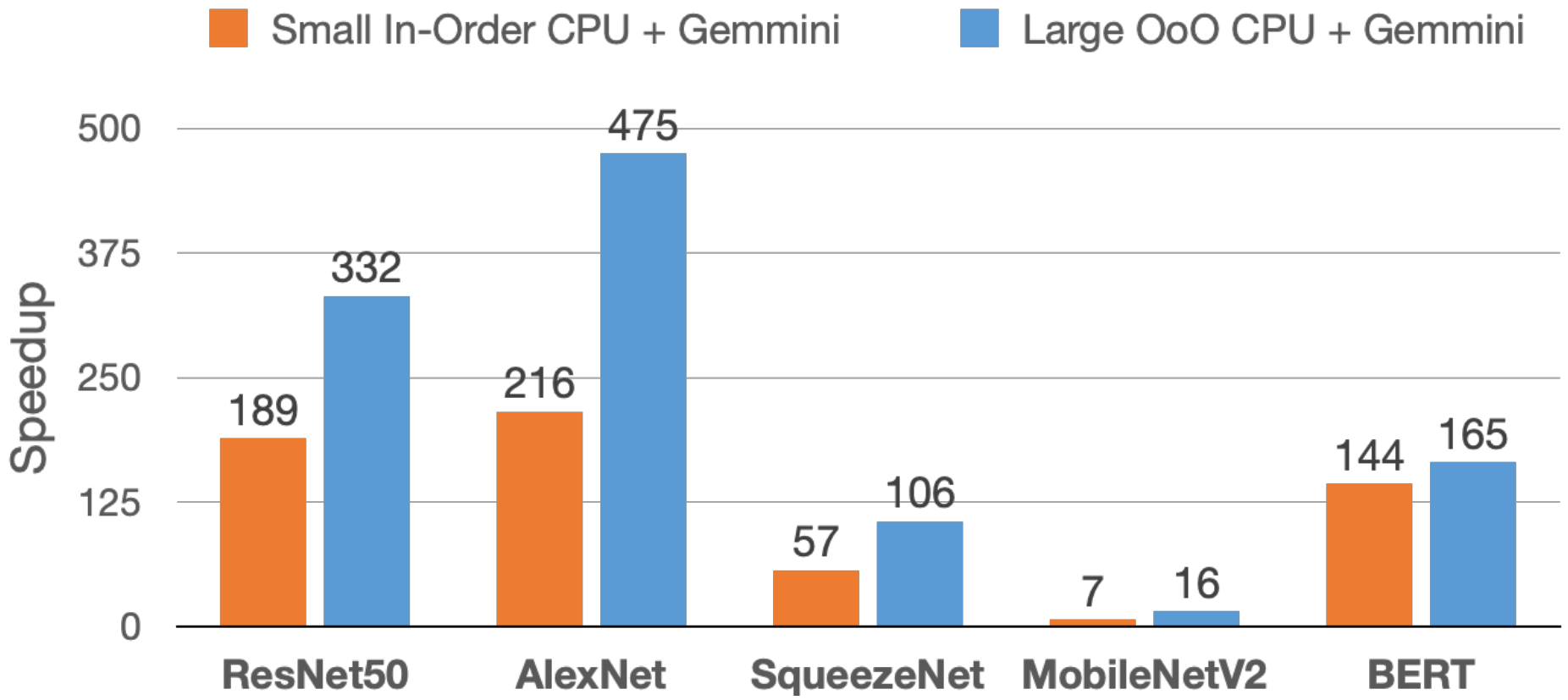
Gemmini: Full SoC



Gemmini: Programming Model

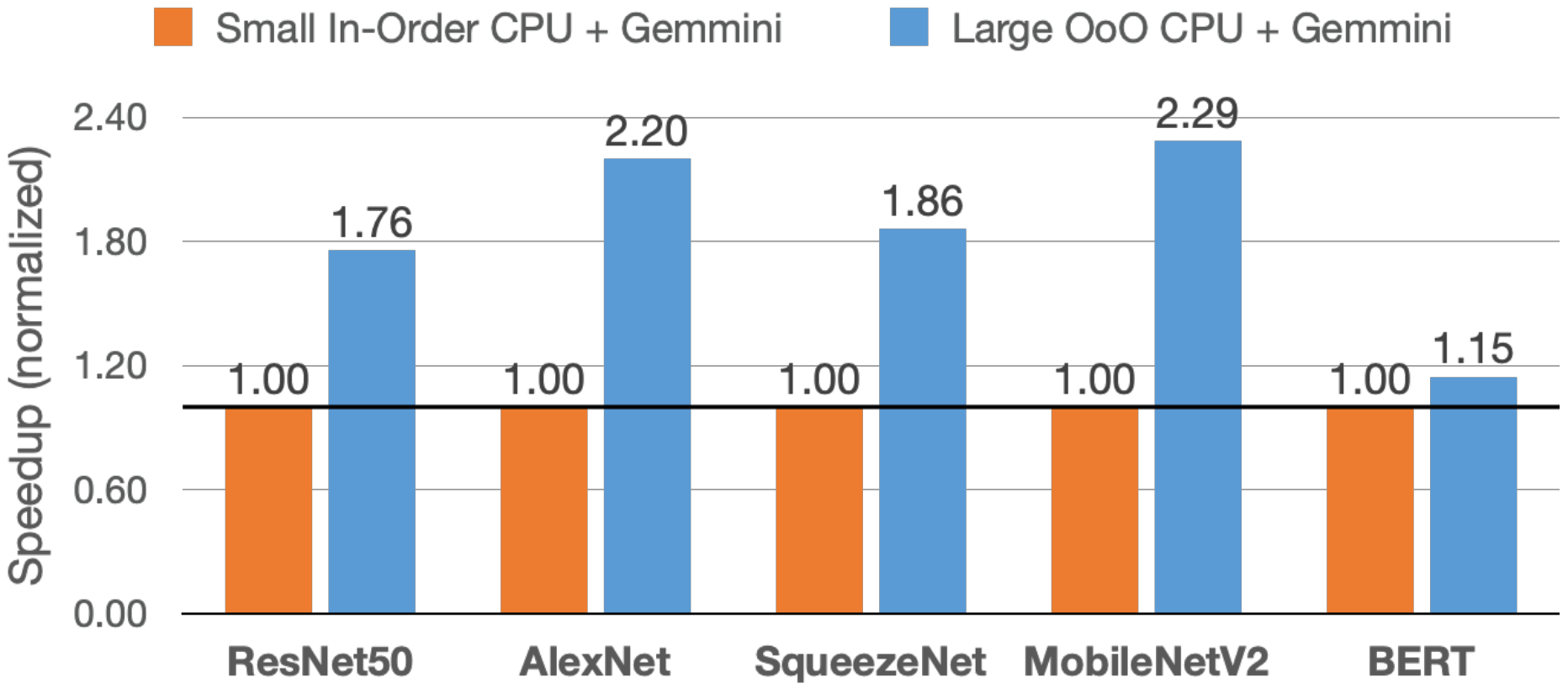
Performance: Evaluating Host CPUs

- "Im2col" runs on CPU, matmuls run on Gemmini



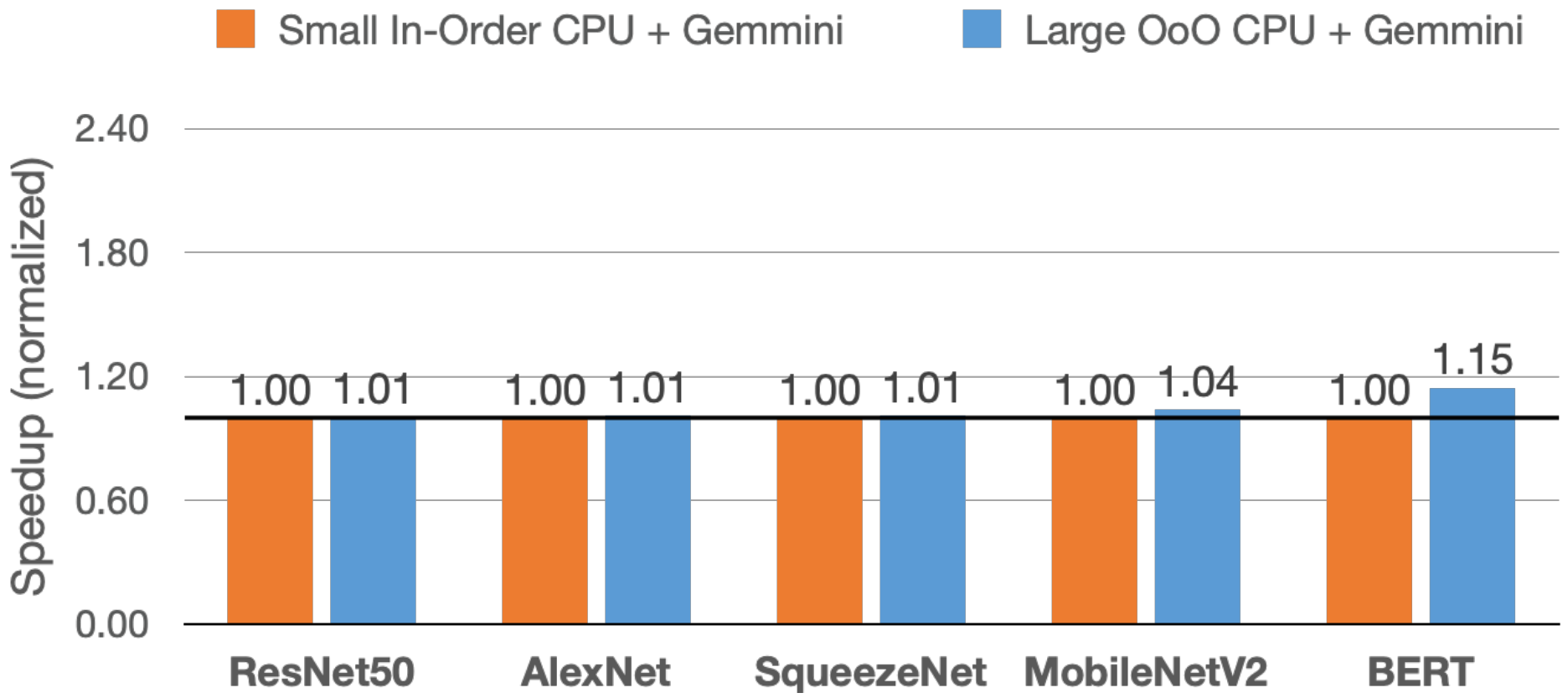
Performance: Evaluating Host CPUs

- "Im2col" runs on CPU, matmuls run on Gemmini



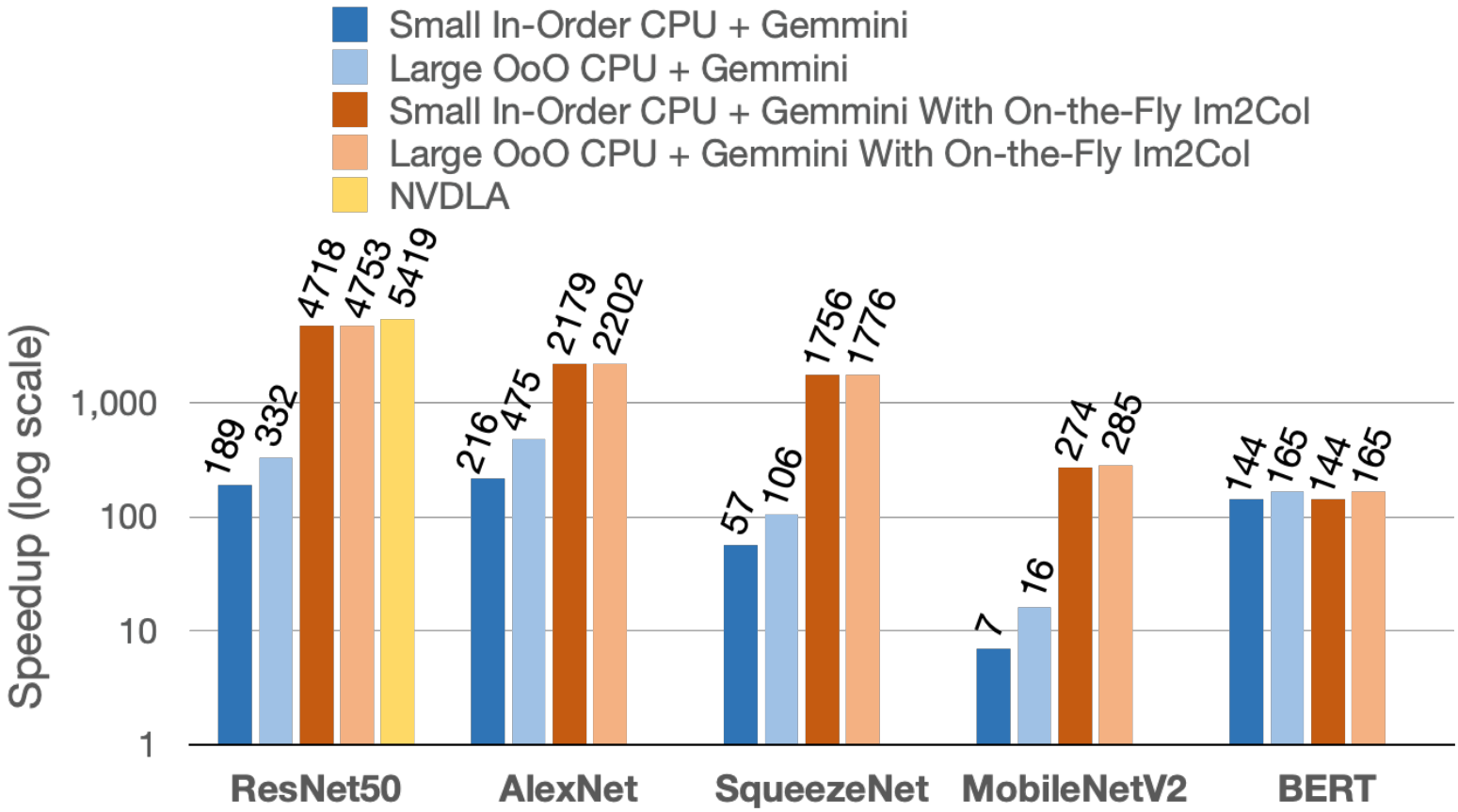
Performance: Evaluating Optional Functional Units

- "Im2col" and matmuls both run on Gemmini



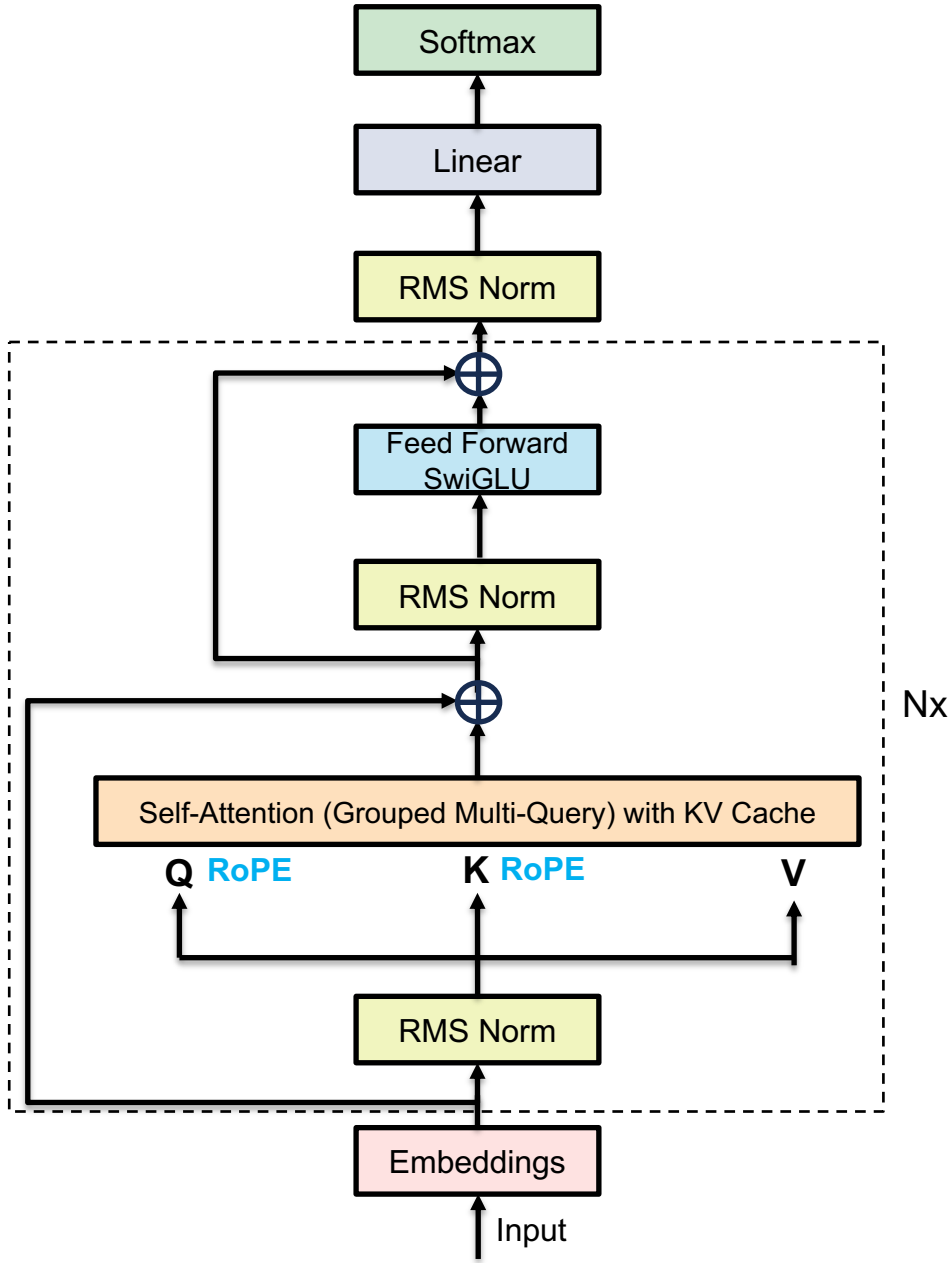
Performance: Overall

- Inference speed:
  - ResNet50: 40.3 FPS
  - AlexNet: 79.3 FPS
  - MobileNetV2: 37.5 FPS
  - BERT: 165x speedup
- About 80% as fast as NVDLA



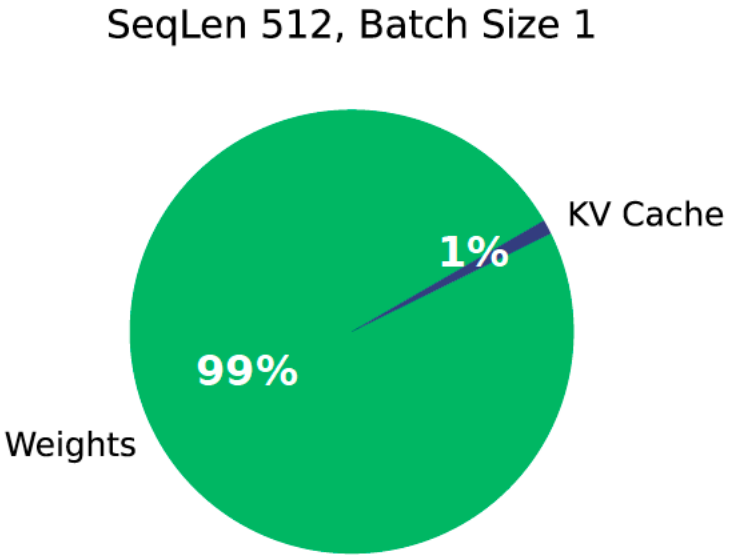
LLaMA-7B inference

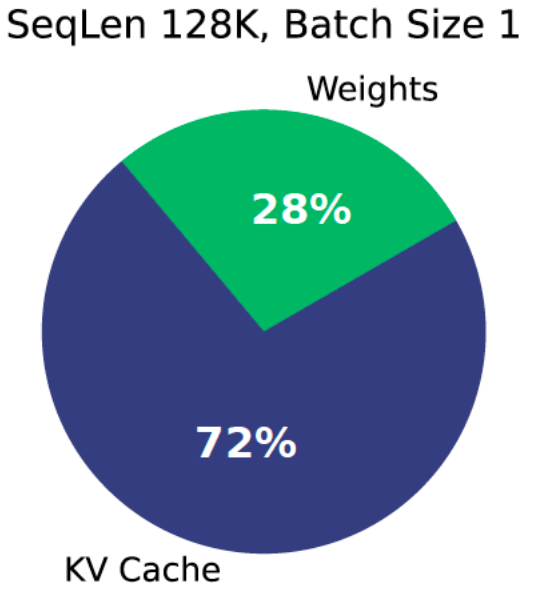
- Matrix-matrix multiplication between weights and activation during QKV generation
- Matrix-vector multiplication between activations for attention mechanism and KV cache
- Element-wise add/mult
- Non-linear operations
  - Rotatory positional encoding (RoPE)
  - RMS Norm
  - SwiGLU
  - Softmax
  - Divide by constant for attention scores



LLaMA-7B Memory requirements

- Depending on the sequence length (SeqLen) the memory footprint changes
- For low SeqLen, weights are the primary memory hogs
- Increasing SeqLen, shifts the bottleneck to KV cache storage
- Most tinyML devices will operate on low SeqLen
- Primary focus is to compress weights





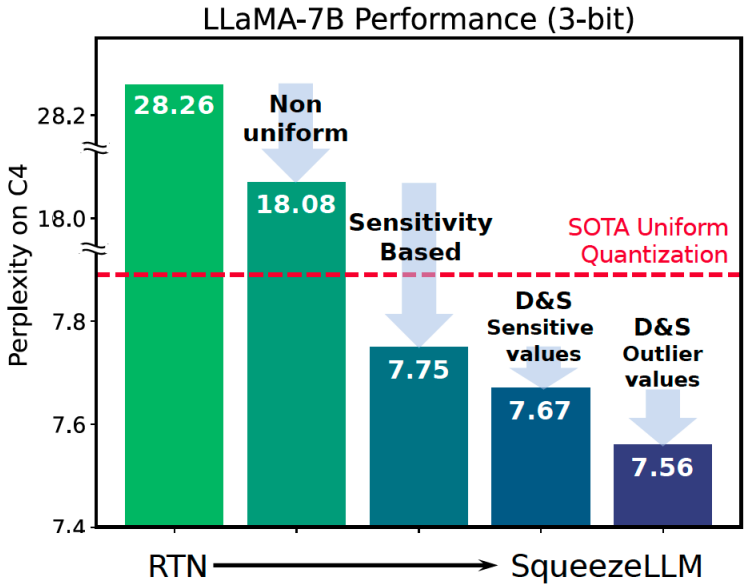
Hooper et al. KVQuant, arxiv, 2024

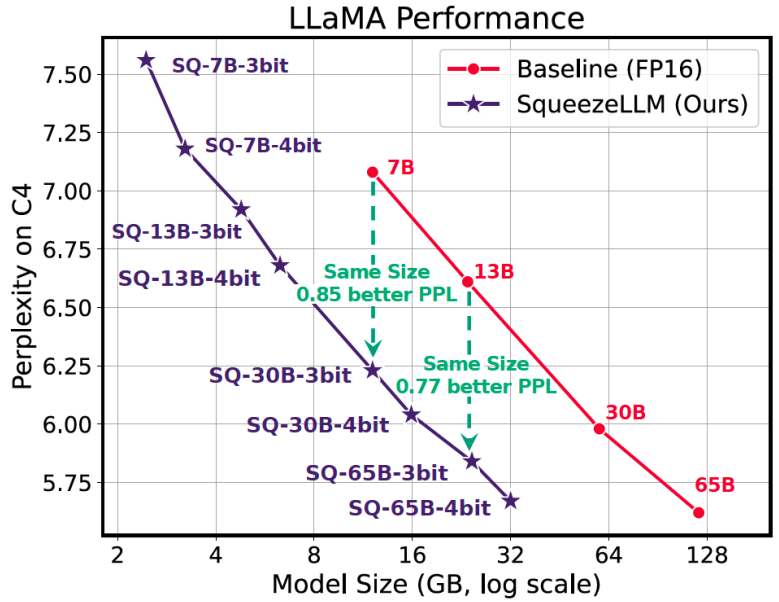
SqueezeLLM

Two key approaches

- Sensitivity-based non-uniform quantization → quantization bins are allocated closer to sensitive values
- Dense and sparse decomposition to retain both sensitive values and outliers as full-precision sparse format

SqueezeLLM on LLaMA-7B

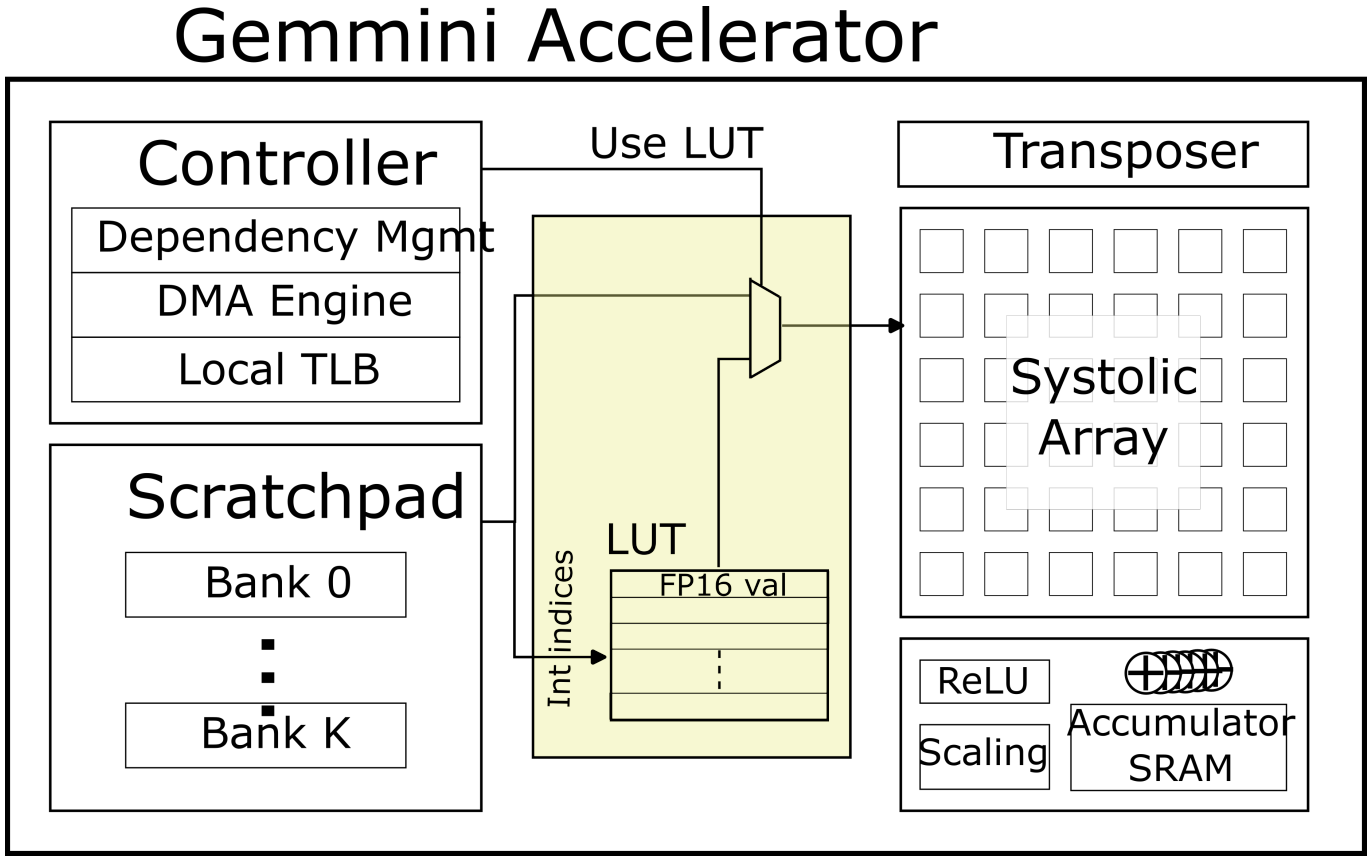




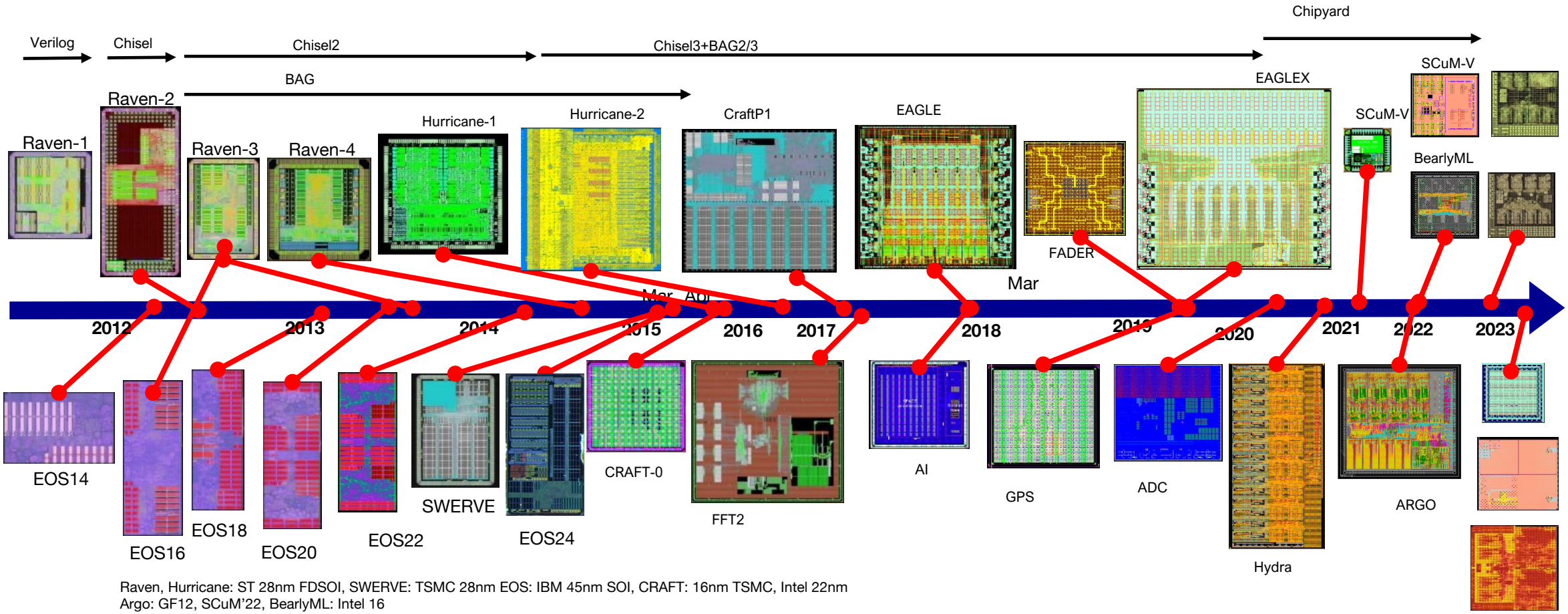
Kim et al. SqueezeLLM, arxiv Feb. 2024

Non-uniform quantization on Gemmini

- Int2/3/4 indices to fp16 lookup table
- Weights are quantized to Int2/3/4 and stored in memory of Gemmini
- Before MatMul, the Int2/3/4 indices are used to dequantize weights to fp16
- Dense MatMul happens with fp16 in the systolic array
- Results are scaled using per-channel scaling factors



CHIPYARD For Research Chips

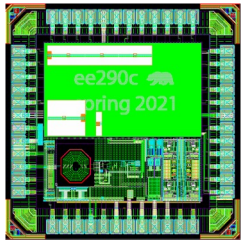


Raven, Hurricane: ST 28nm FDSOI, SWERVE: TSMC 28nm EOS: IBM 45nm SOI, CRAFT: 16nm TSMC, Intel 22nm
Argo: GF12, SCuM'22, BearlyML: Intel 16

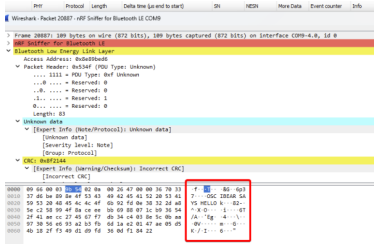
CHIPYARD For Class Tapeouts

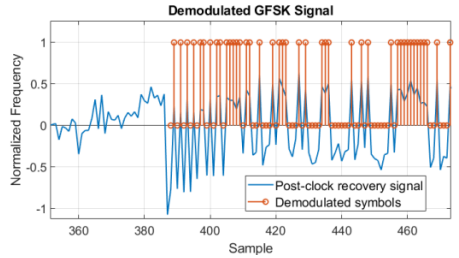
2021:
18 students

TSMC 28nm
1mm x 1mm



OSCIBear: 32b RISC-V + BLE + AES + Power



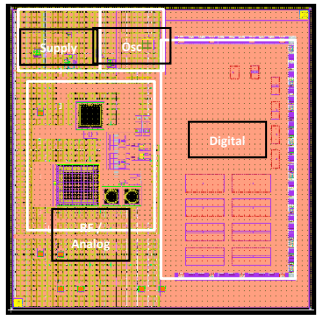


Processor, transmitter functional

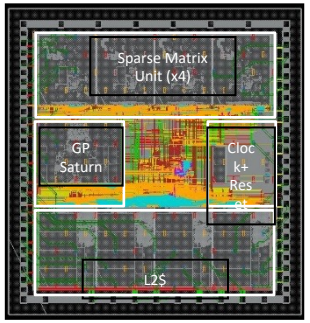
HotChips'23

2022:
41 students

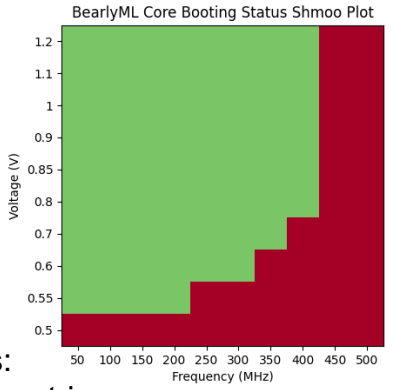
2x Intel 16
2mm x 2mm



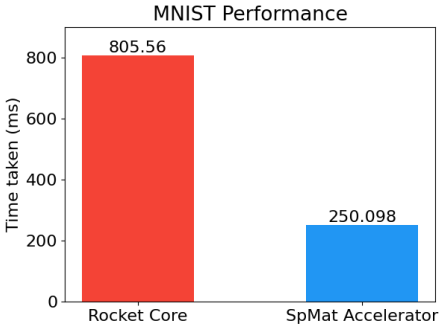
SCuM-V'22: 64b RISC-V core, BLE + 802.15.4, LDOs, references



BearlyML'22: 5 RISC-V cores:
4 Rocket with custom sparse matrix acc,
Saturn-V, NoC, PLL, L2



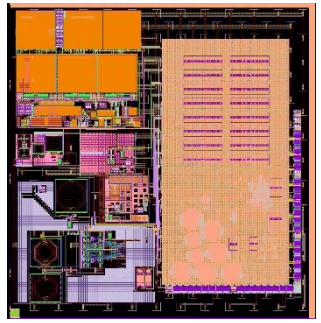
Shmoo plot, running MNIST



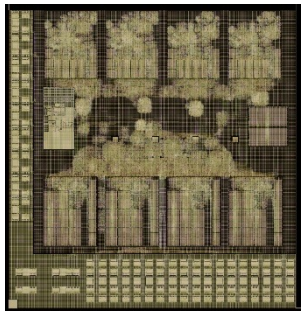
HotChips'23

2023:
54 students

3x Intel 16
2mm x 2mm

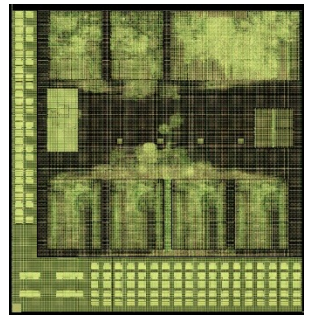


SCuM-V'23: 32b RISC-V core, BLE + 802.15.4, LDOs, references, radar



BearlyML'23: 4 RISC-V Rockets with custom sparse matrix acc, near-memory acc, NoC, L2$

2024:
70 students



RoboChip'23: 2 RISC-V Rockets with Kalman, LQR acc, BooM + MTE, NoC, L2$

Bedtime Story Demo on BeralyML’23

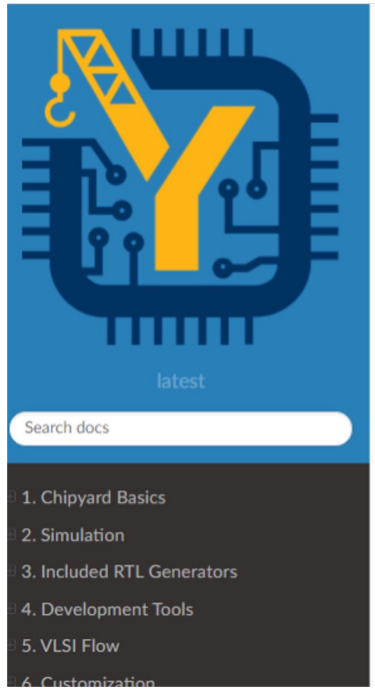
Running a small LLaMA model trained on tinyStories dataset…

CHIPYARD

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

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

Thanks to all internal/external Chipyard developers



https://github.com/ucb-bar/chipyard

https://chipyard.readthedocs.io/en/stable/

Copyright Notice

www.tinyml.org