

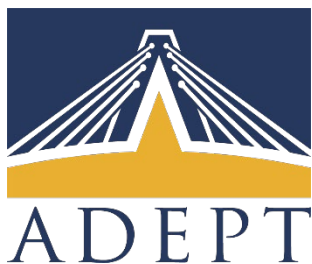


Gemmini: Enabling Systematic Deep-Learning Architecture Evaluation via Full-Stack Integration



FROM CHIPS TO SYSTEMS — LEARN TODAY, CREATE TOMORROW

Hasan Genc, Seah Kim, Alon Amid, Ameer Haj-Ali, Vighnesh Iyer, Pranav Prakash, Jerry Zhao, Daniel Grubb, Harrison Liew, Howard Mao, Albert Ou, Colin Schmidt, Samuel Steffl, John Wright, Ion Stoica, Jonathan Ragan-Kelley, Krste Asanovic, Borivoje Nikolic, Yakun Sophia Shao





DNNs are exploding in popularity...



Matt Christenson/BLM/2017



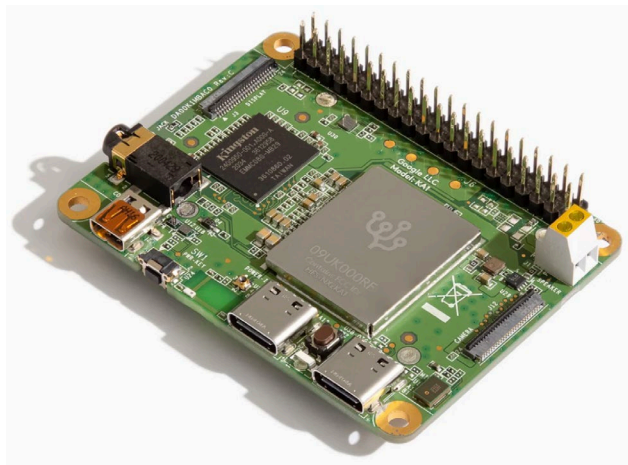
*By Dllu - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=64517567>*



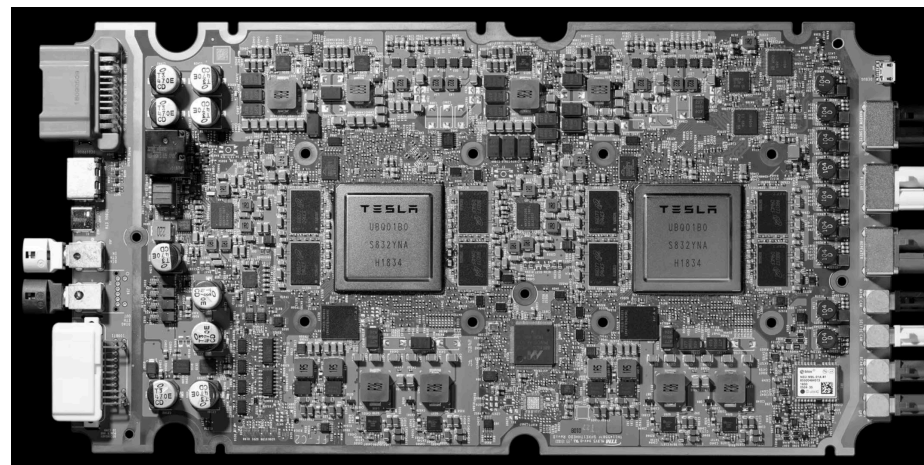
Apple Support



Which means DNN **ACCELERATORS**
are exploding in popularity...



Edge TPU



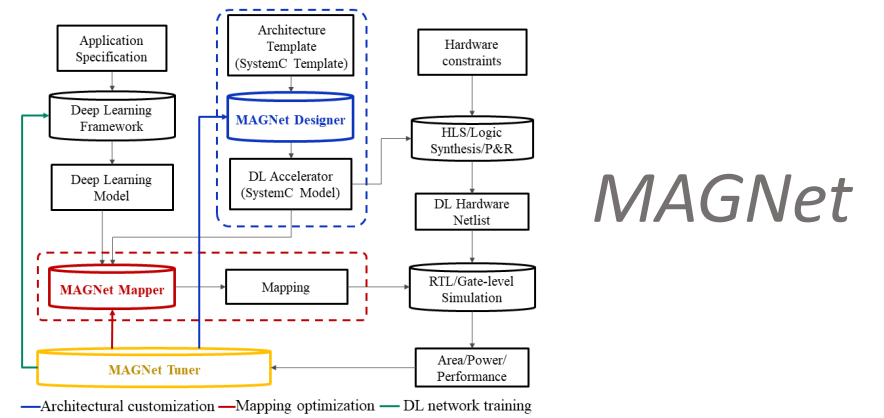
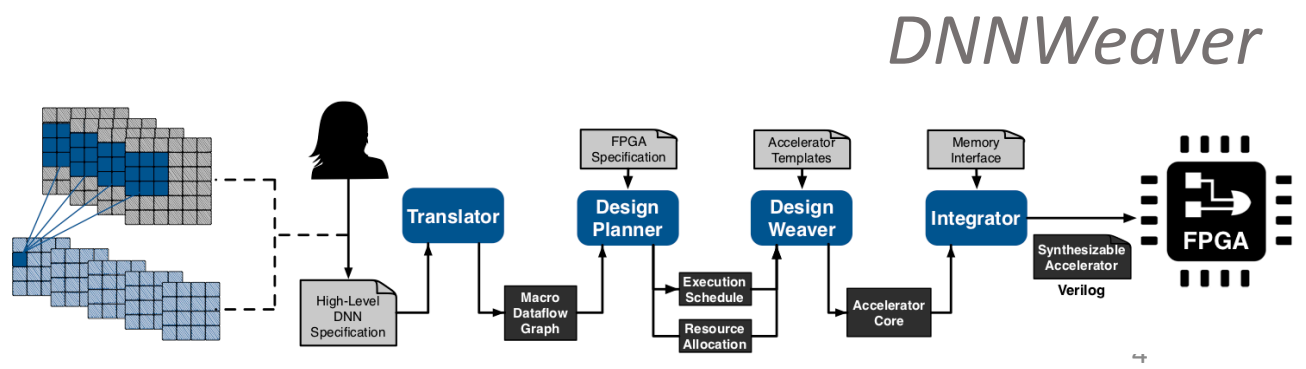
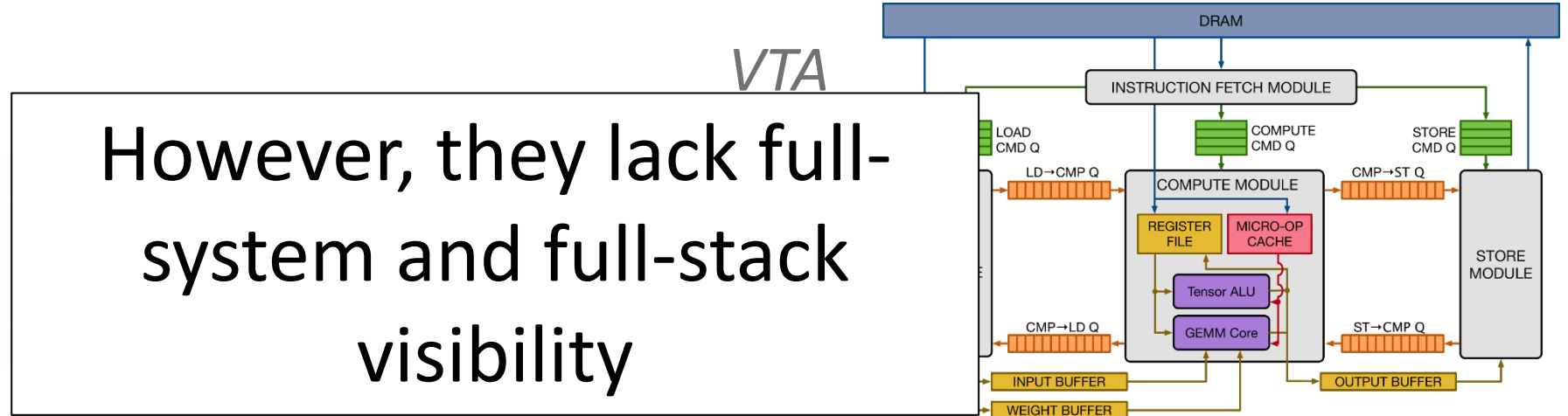
Tesla FSD



Cloud TPU



Which means DNN accelerator
GENERATORS are exploding in popularity...



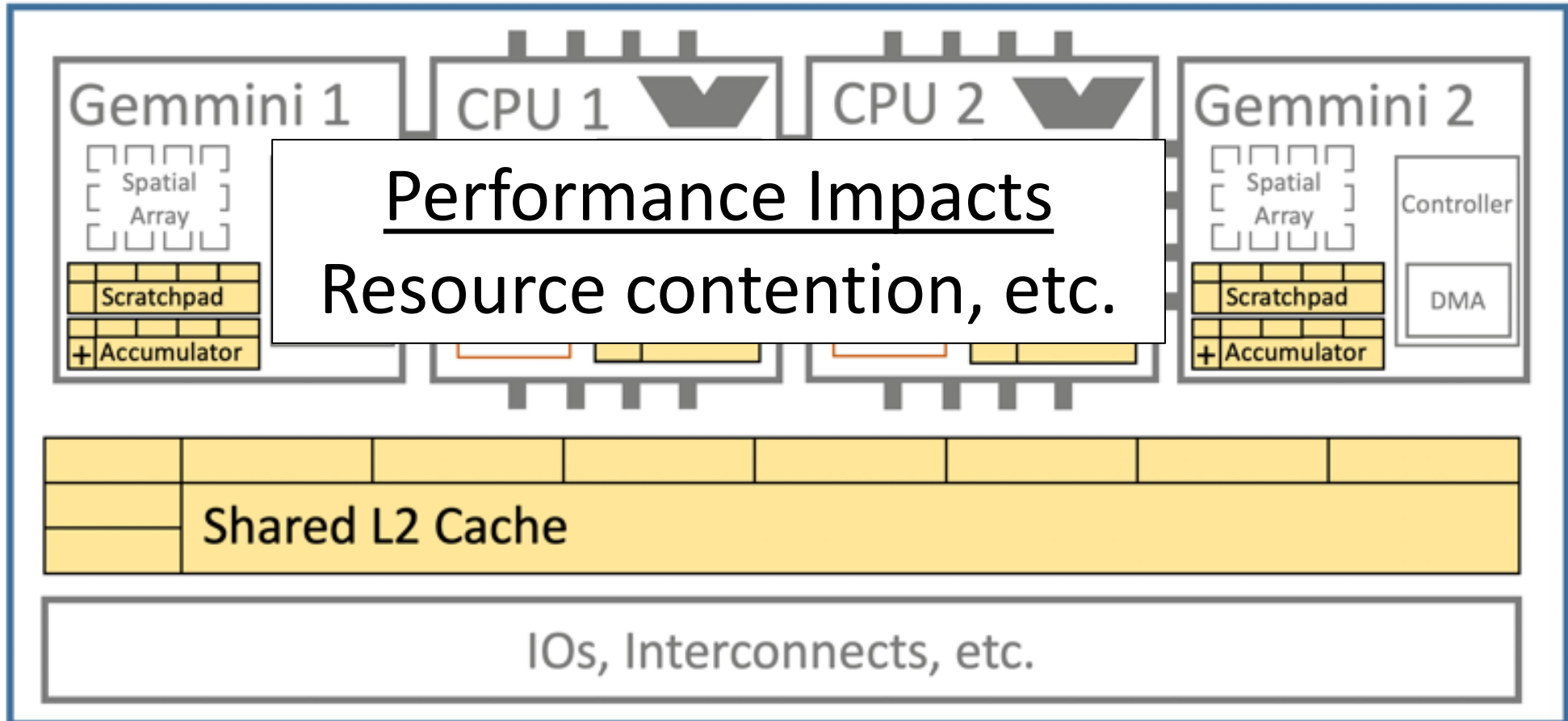


Full-System Visibility



Full-System Visibility: SoC

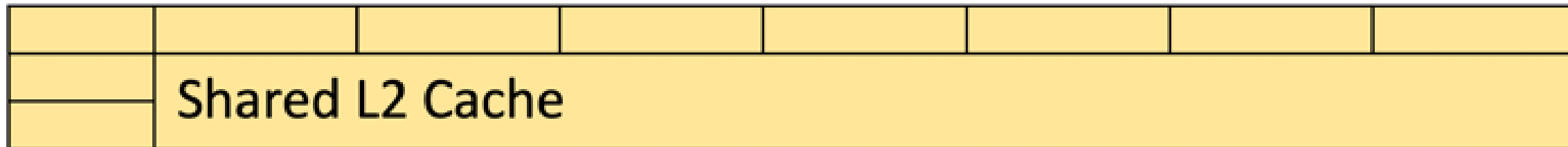
SoC





Full-System Visibility: Memory Hierarchy

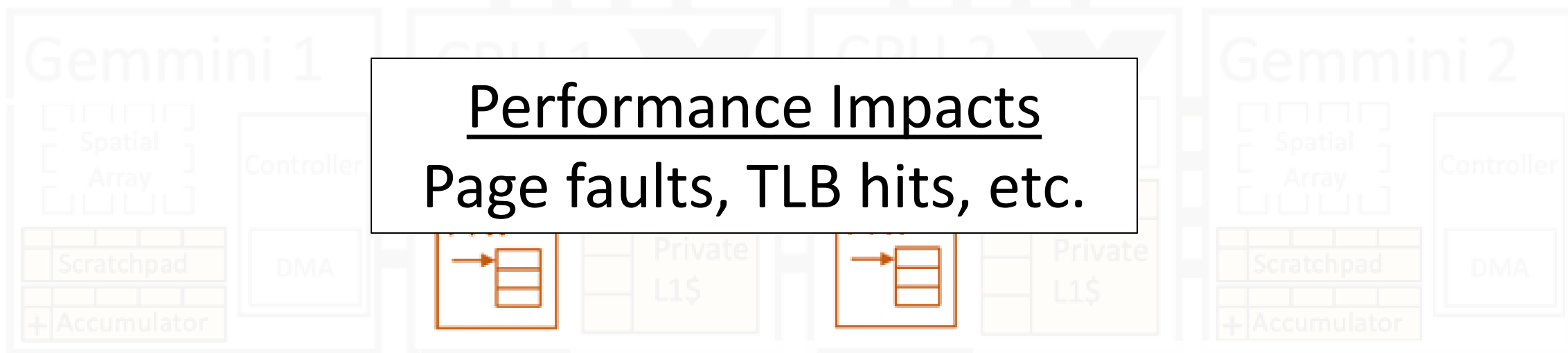
Performance Impacts
Cache coherence, miss rates/latencies, etc.





Full-System Visibility: Virtual Addresses

Performance Impacts
Page faults, TLB hits, etc.

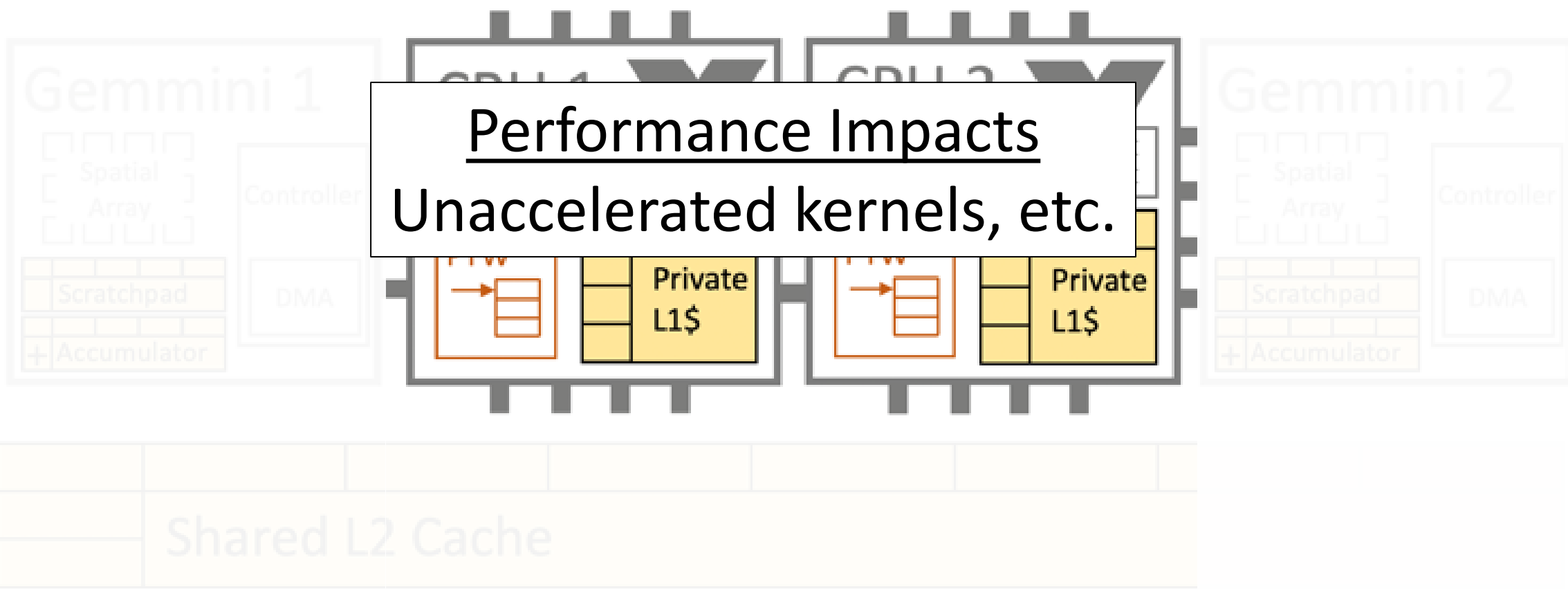


I/Os, Interconnects, etc.



Full-System Visibility: Host CPUs

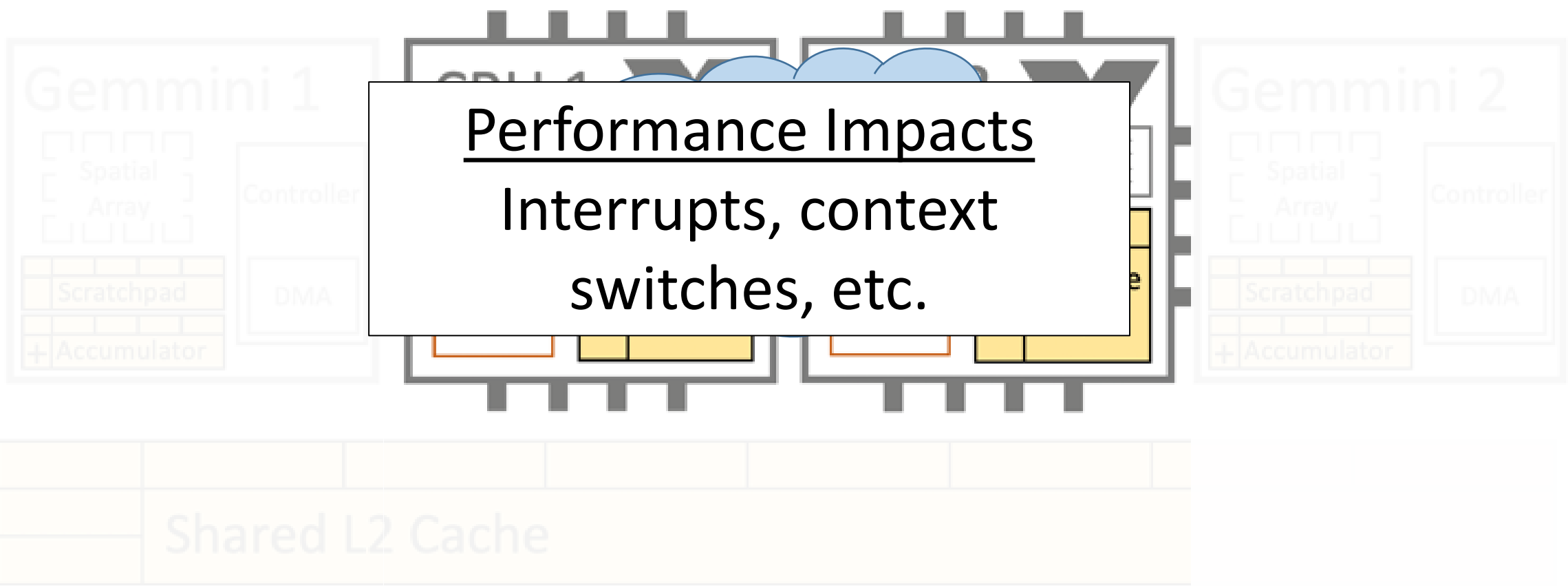
Performance Impacts
Unaccelerated kernels, etc.





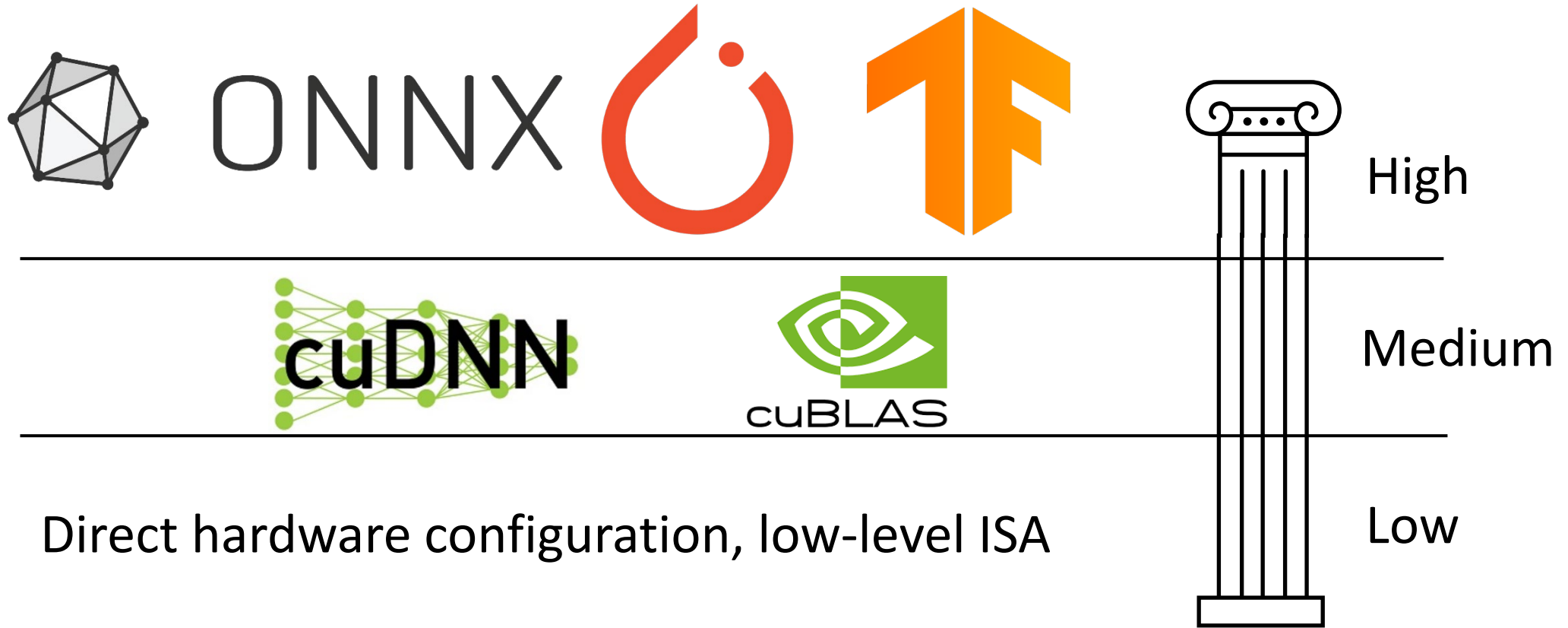
Full-System Visibility: Operating System

Performance Impacts
Interrupts, context
switches, etc.





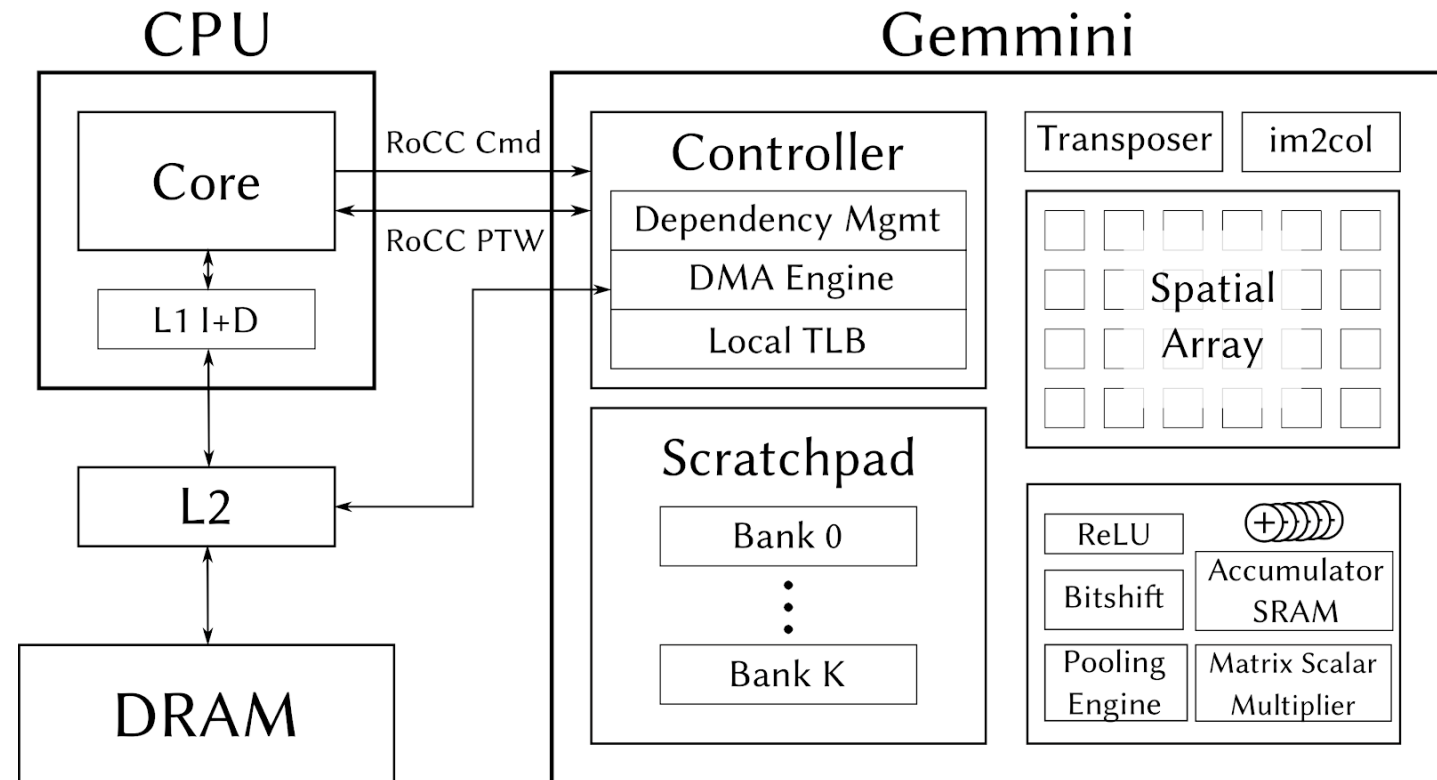
Full-Stack Visibility





Gemmini

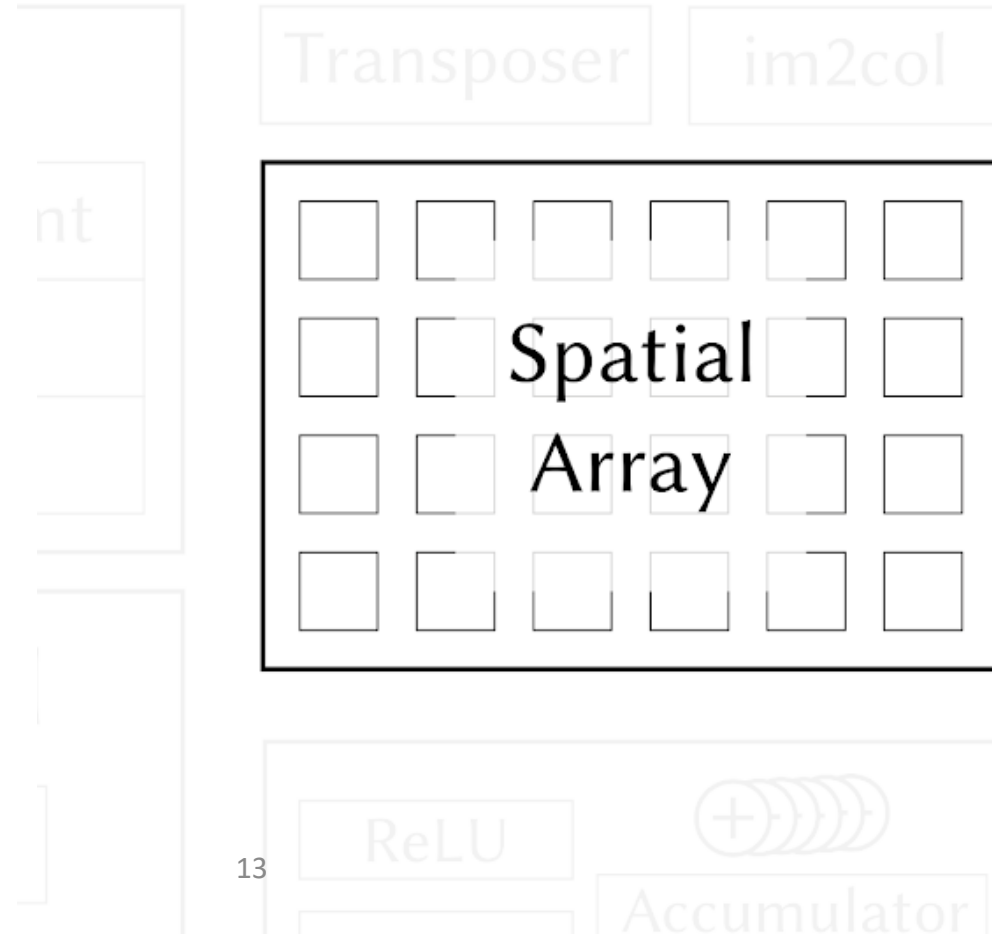
- DNN accelerator generator
- Flexible hardware template
- Full-stack
- Full-system





Gemmini: Spatial Array

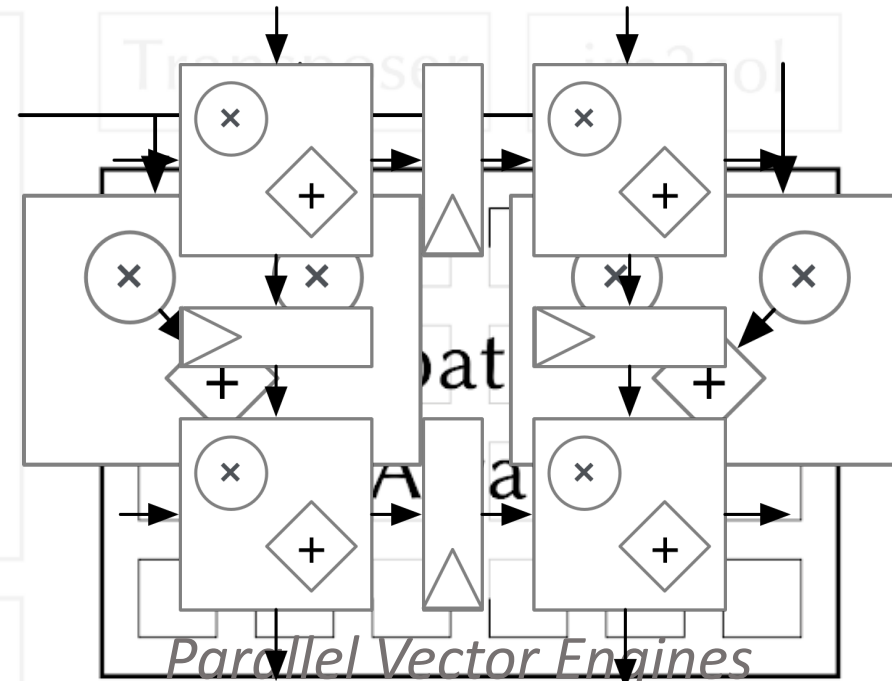
- Parameters:
 - Dataflow
 - Dimensions
 - Pipelining





Gemmini: Spatial Array

- Parameters:
 - Dataflow
 - Dimensions
 - Pipelining

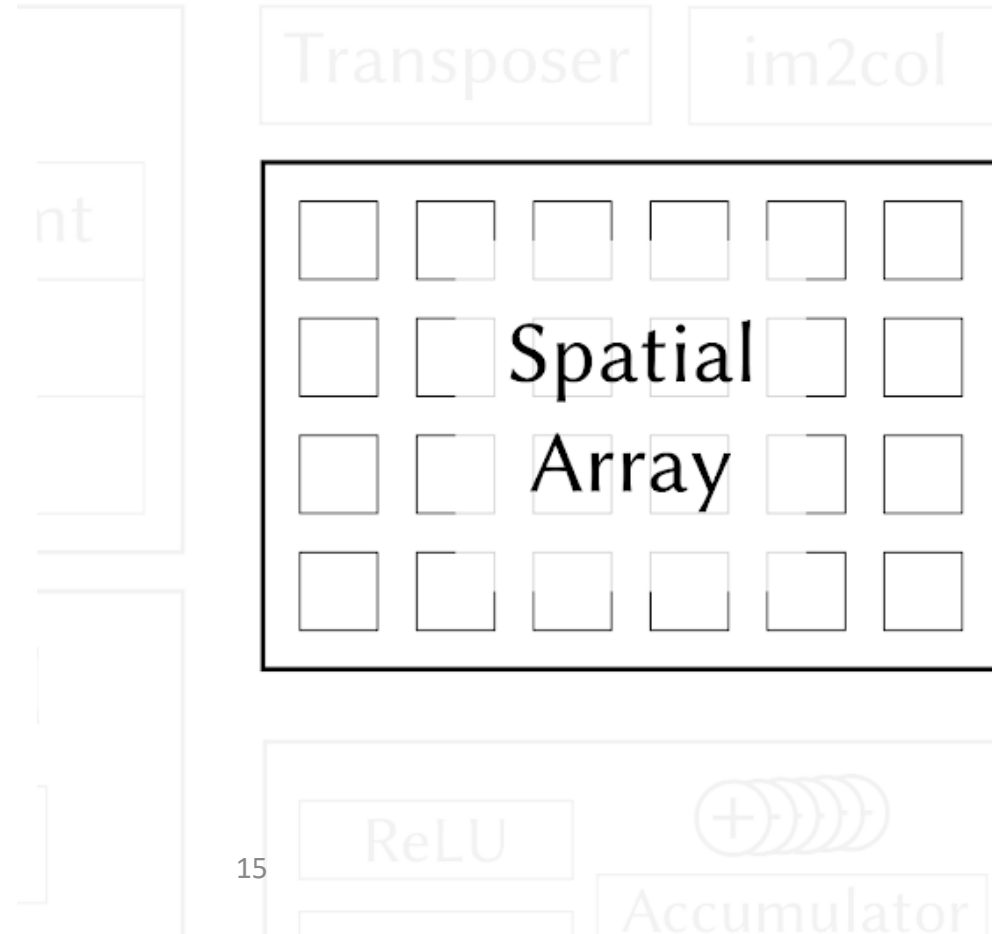


Systolic Array (Like TPU)



Gemmini: Spatial Array

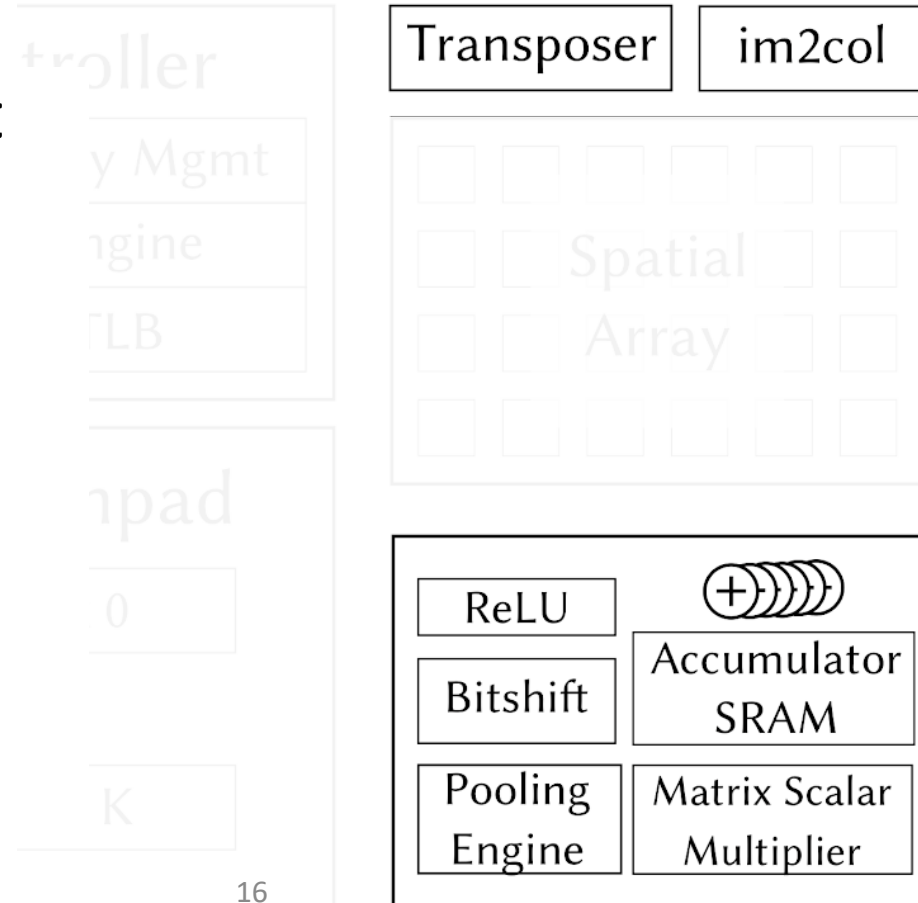
- Parameters:
 - Dataflow
 - Dimensions
 - Pipelining





Gemmini: Non-GEMM Functionality

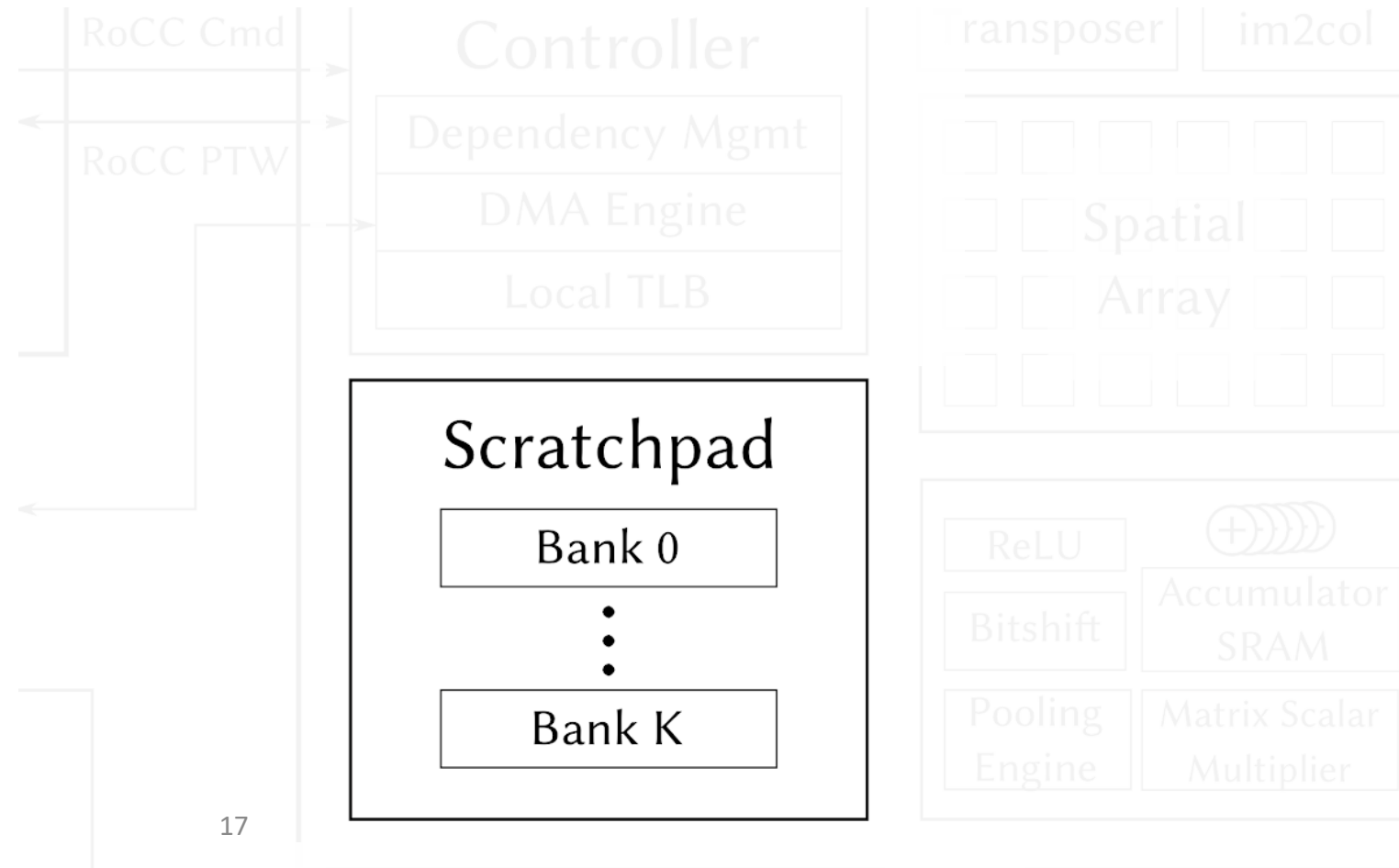
- Can be optimized out at elaboration-time





Gemmini: Local Scratchpad

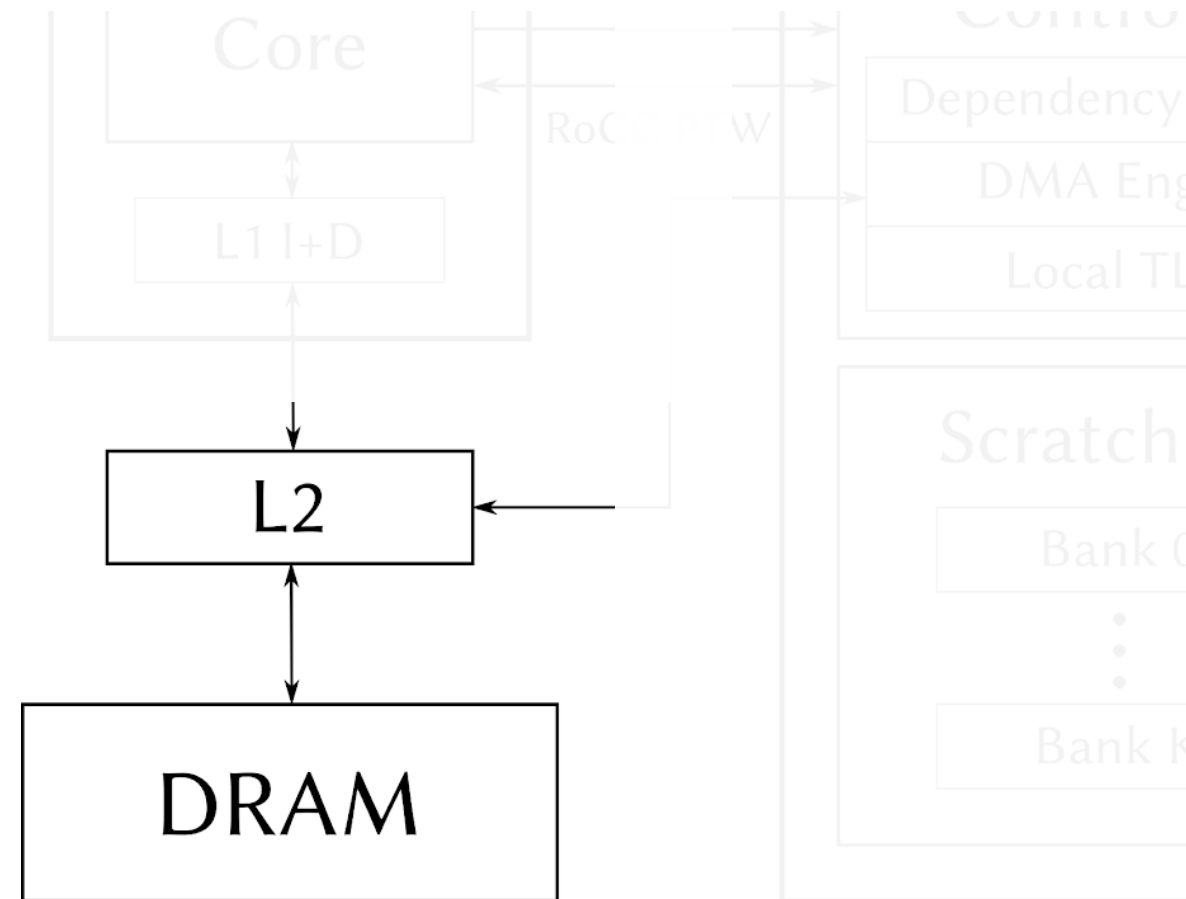
- Parameters:
 - Capacity
 - Banks
 - Single- or dual-port





Gemmini: Global Memory

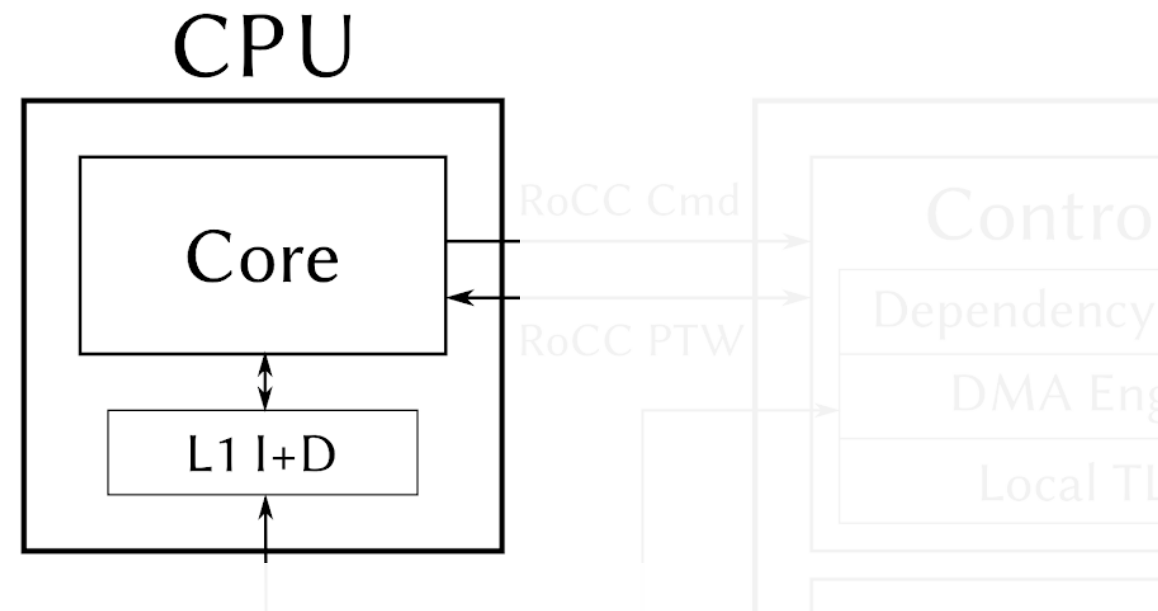
- Parameters:
 - Capacity
 - Banks
 - DRAM controller





Gemmini: Host CPU

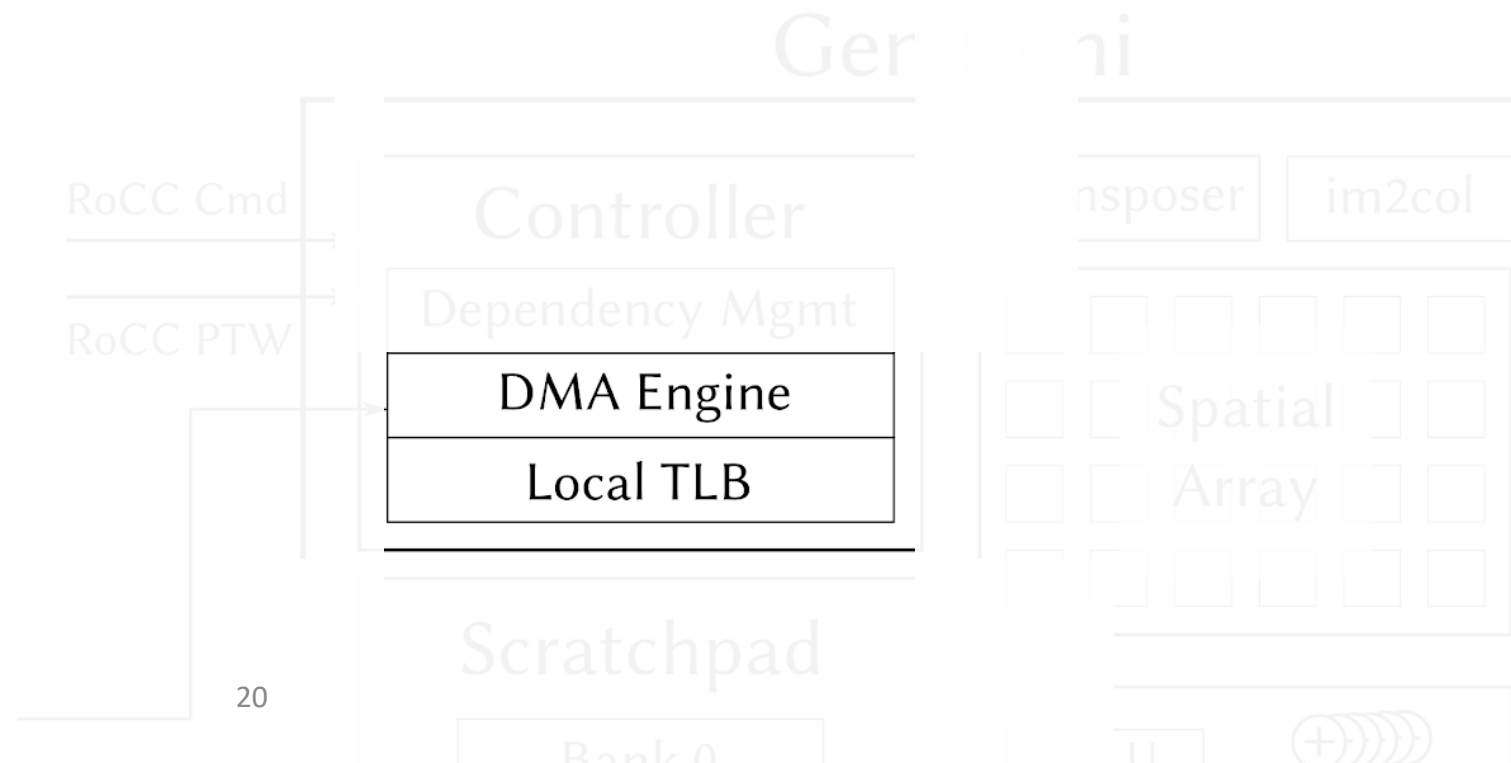
- Parameters:
 - In-order/out-of-order
 - ROB capacity
 - L1 capacity
 - Branch predictor





Gemmini: Virtual Address Translation

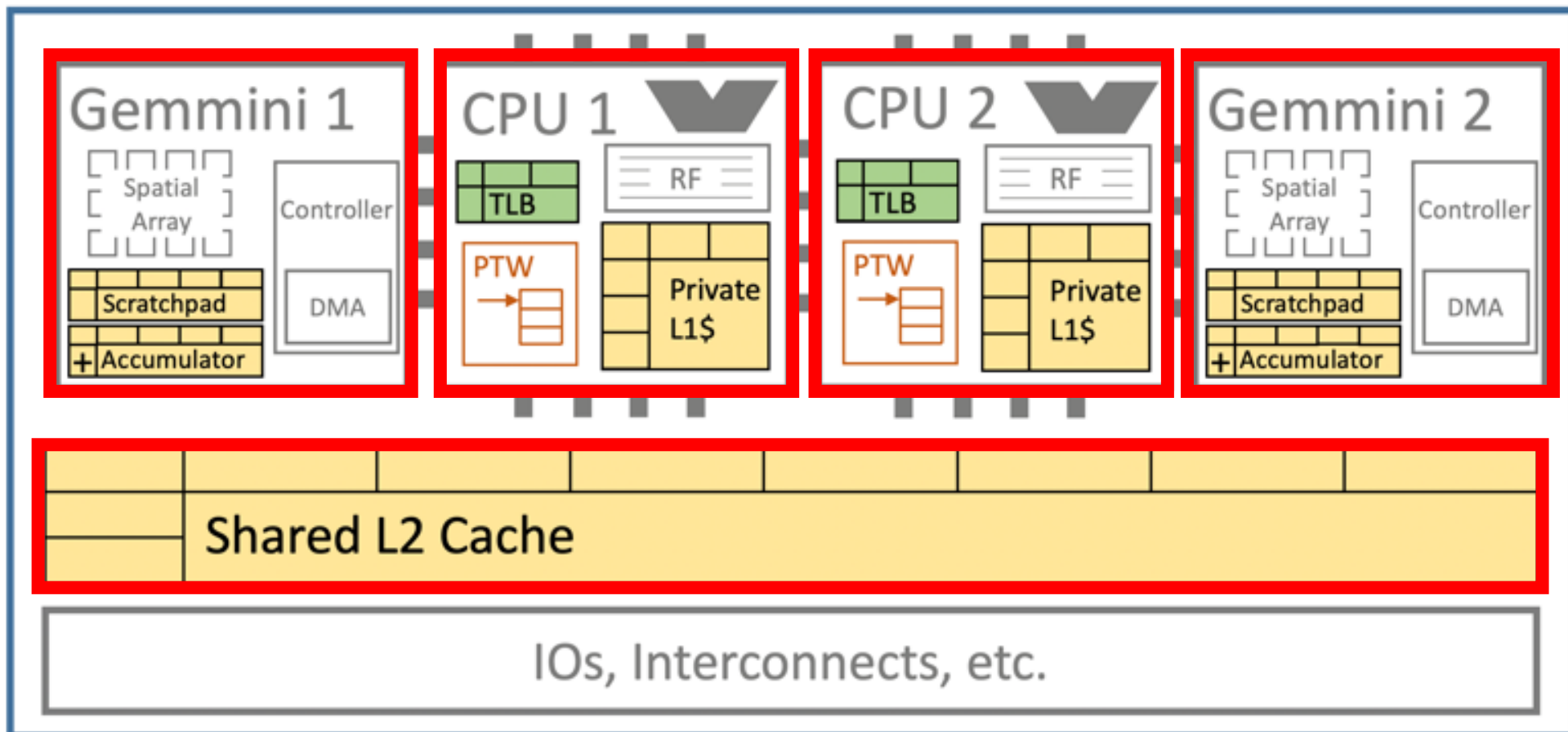
- Parameters:
 - TLB capacity
 - TLB hierarchy
 - e.g. L2 TLB





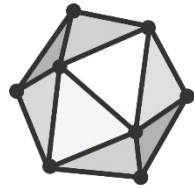
Gemmini: Full SoC

SoC





Gemmini: Programming Model



ONNX



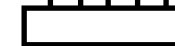
High

matmul(...); conv(...); residual_add(...); max_pool(...);
Hand-tuned C library for DNNs
global_averaging(...)

Medium

Direct hardware configuration, low-level ISA
configure_loads(...); configure_stores(...);
preload_spatial_array(...); feed_spatial_array(...)

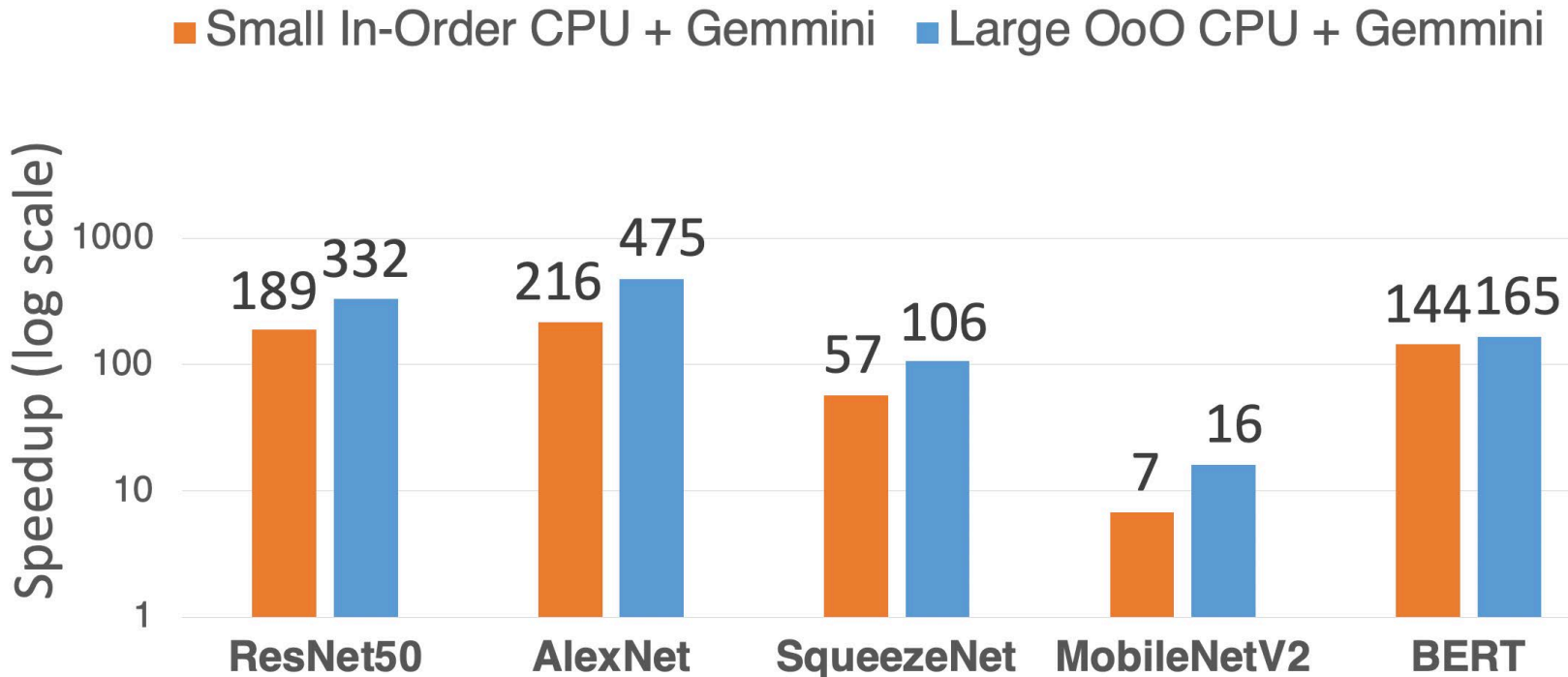
Low





Performance: Evaluating Host CPUs

- “Im2col” runs on CPU, matmuls run on Gemmini

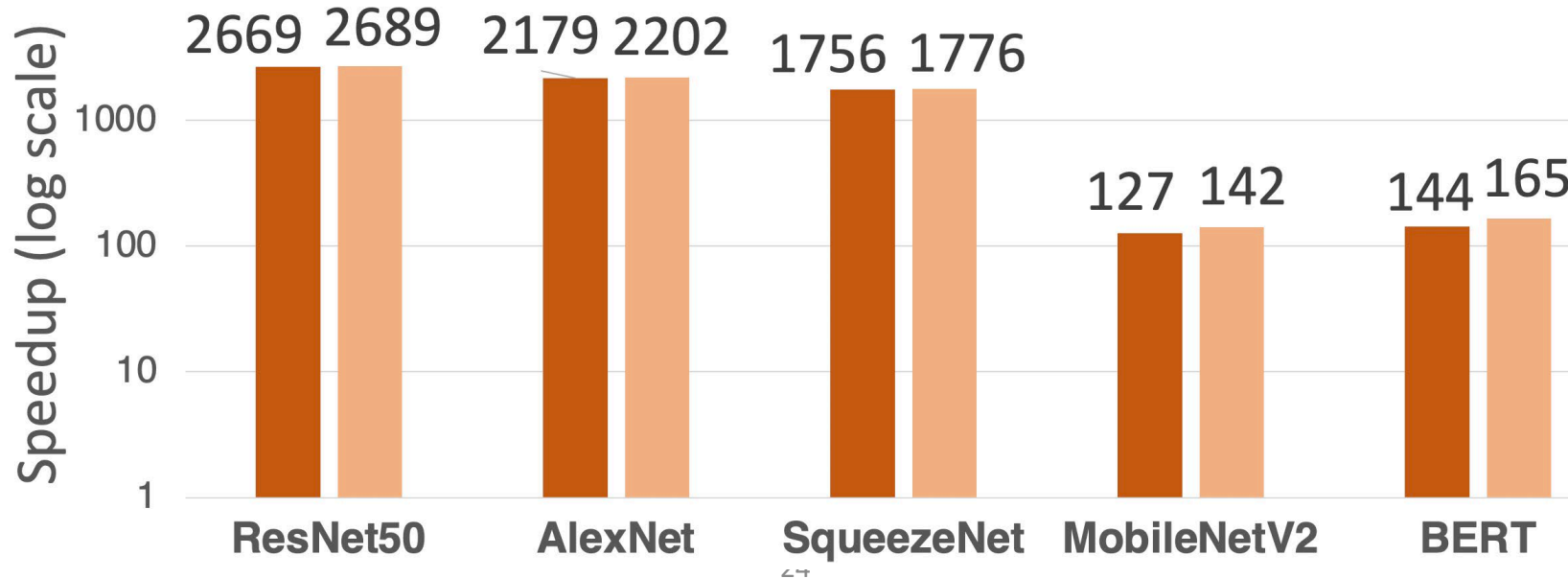




Performance: Evaluating Optional Functional Units

- “Im2col” and matmuls both run on Gemmini

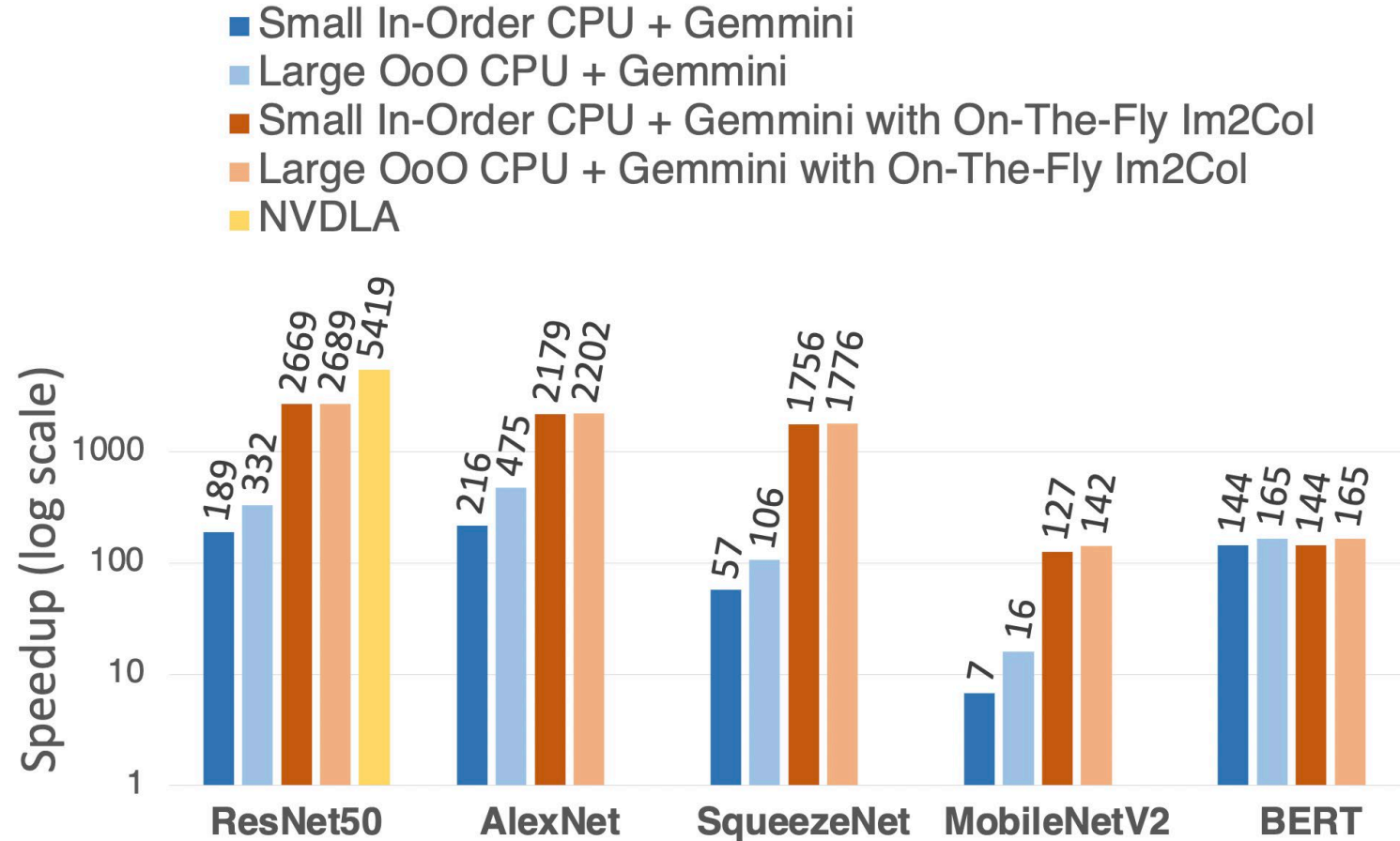
- Small In-Order CPU + Gemmini with On-The-Fly Im2Col
- Large OoO CPU + Gemmini with On-The-Fly Im2Col





Performance: Overall

- DNNs:
 - ResNet50: 22.8 FPS
 - AlexNet: 79.3 FPS
 - MobileNet: 18.7 FPS
 - BERT: 167x speedup
- **About 50% as fast as NVDLA**





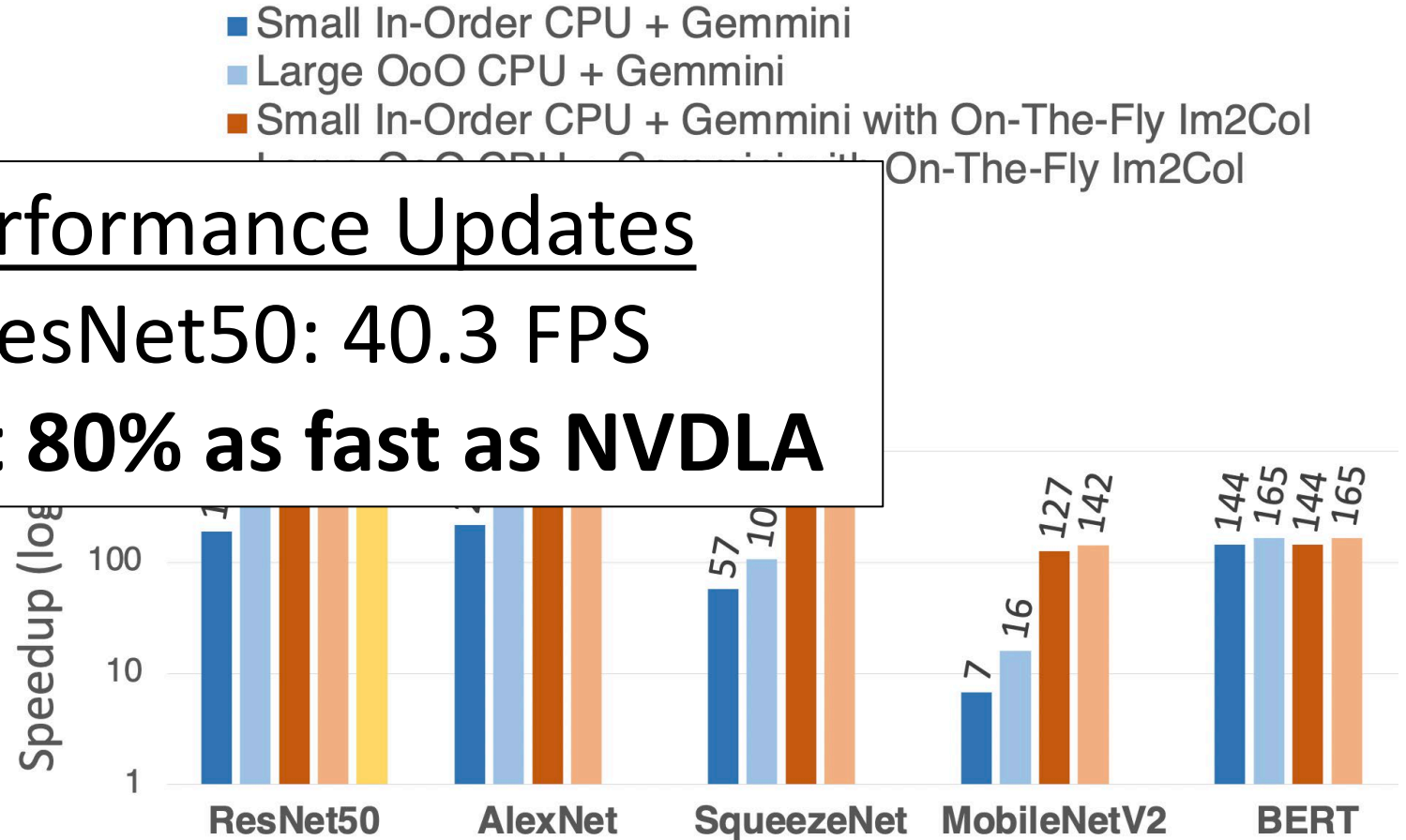
Performance: Overall

• DNNs:

- ~~ResNet50: 22.8 FPS~~
- AlexNet: 79.3 FPS
- MobileNet: 18.7 FPS
- BERT: 167x speedup

• ~~About 50% as fast as NVDLA~~

Performance Updates
 ResNet50: 40.3 FPS
About 80% as fast as NVDLA



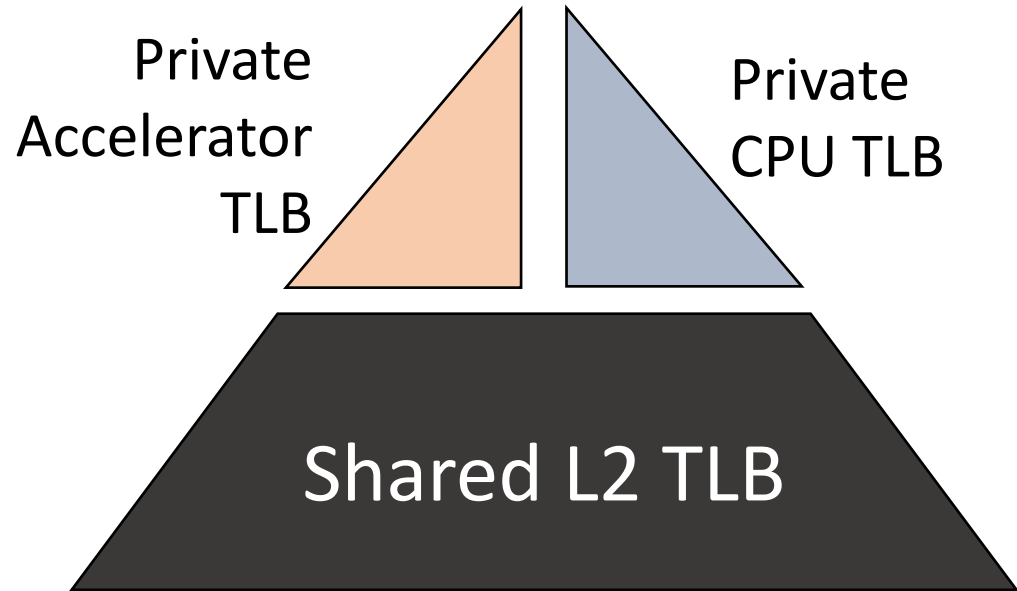


How Does the Full System and Full Stack Affect Performance?

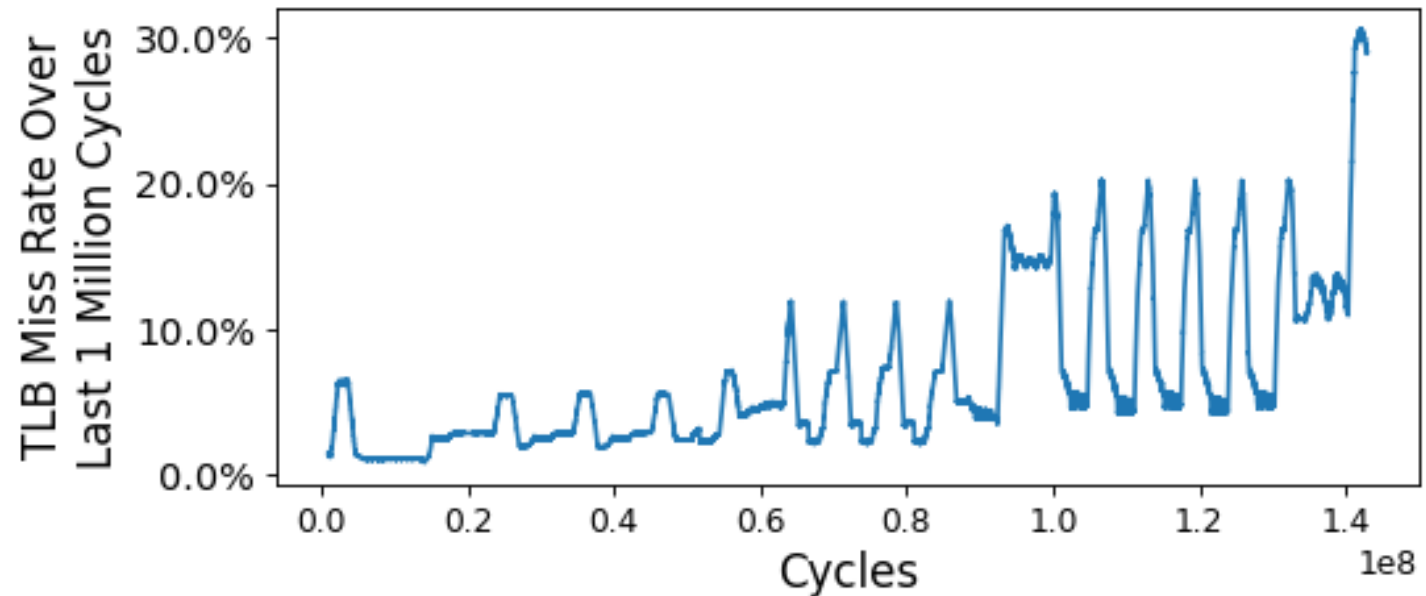


Case Study: Virtual Memory for DNNs

Two-level TLB hierarchy



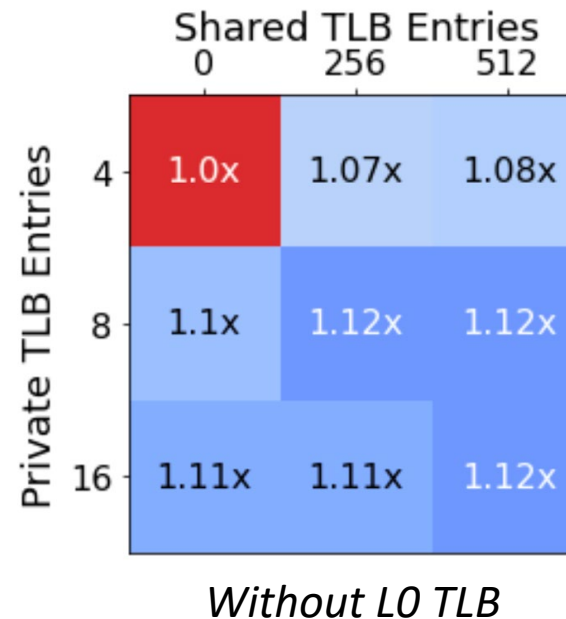
TLB Misses for ResNet50





Case Study: Virtual Memory for DNNs

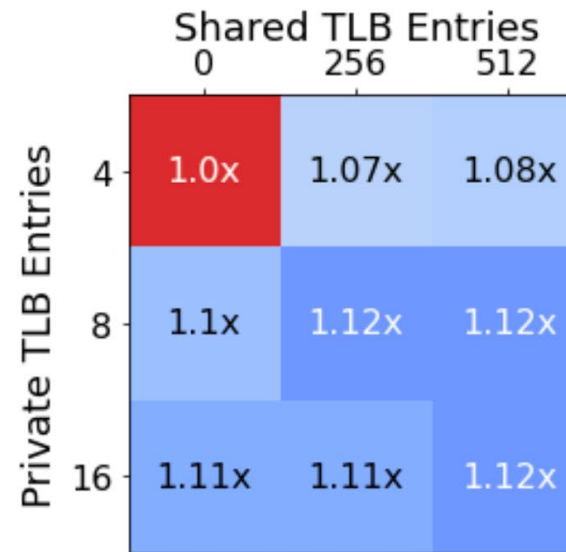
- Small private TLB much more impactful



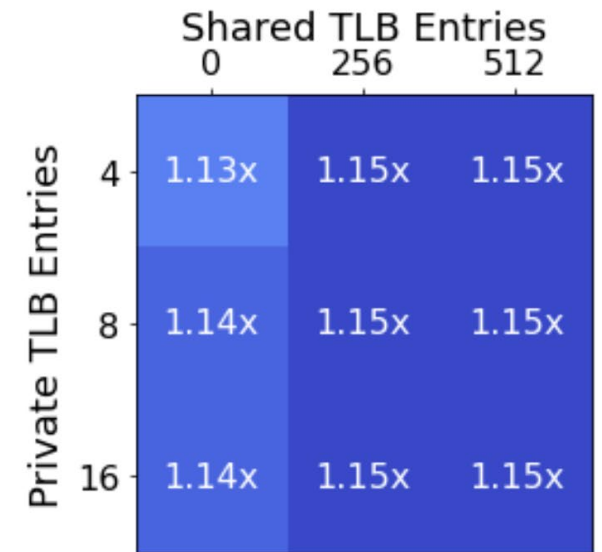


Case Study: Virtual Memory for DNNs

- Small private TLB much more impactful
- Low-cost optimizations:
 - Single-entry L0 TLB filters out consecutive TLB requests to same page



Without L0 TLB

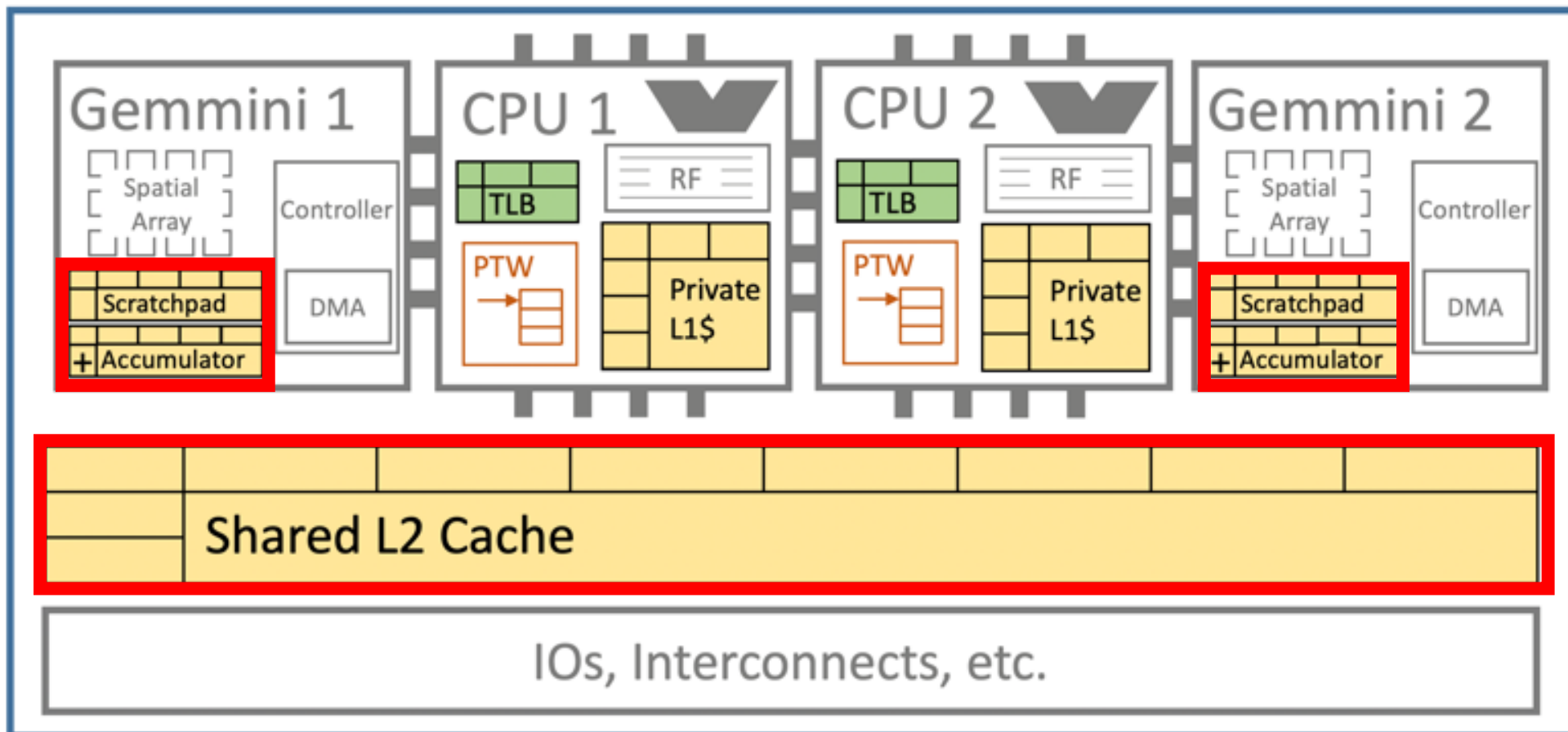


With L0 TLB



Case Study: Memory Partitioning

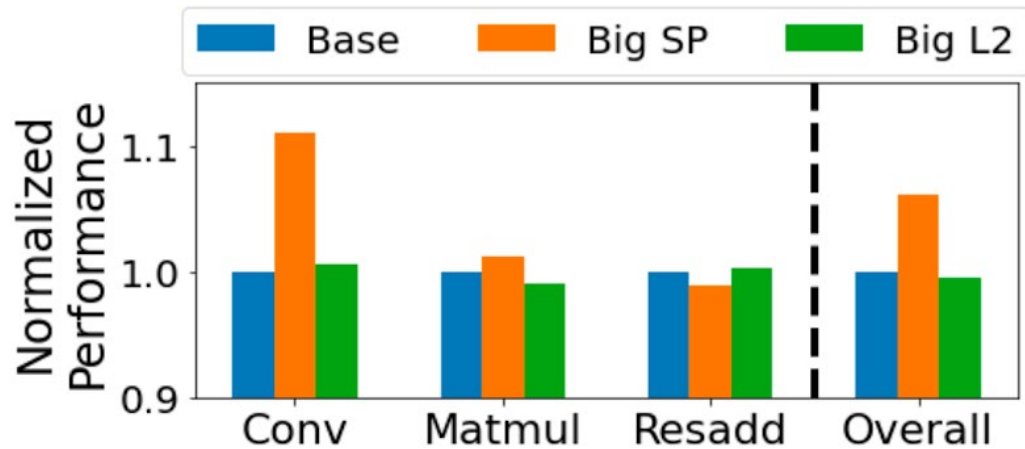
SoC





Case Study: Memory Partitioning

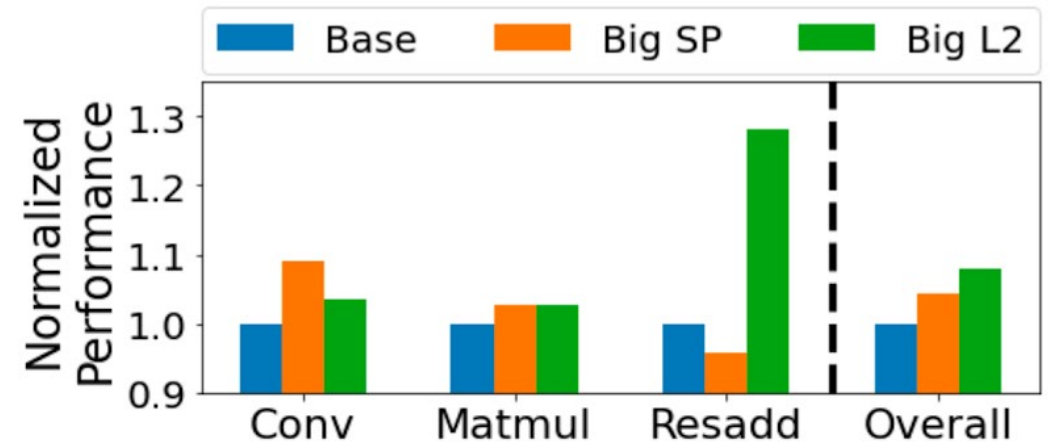
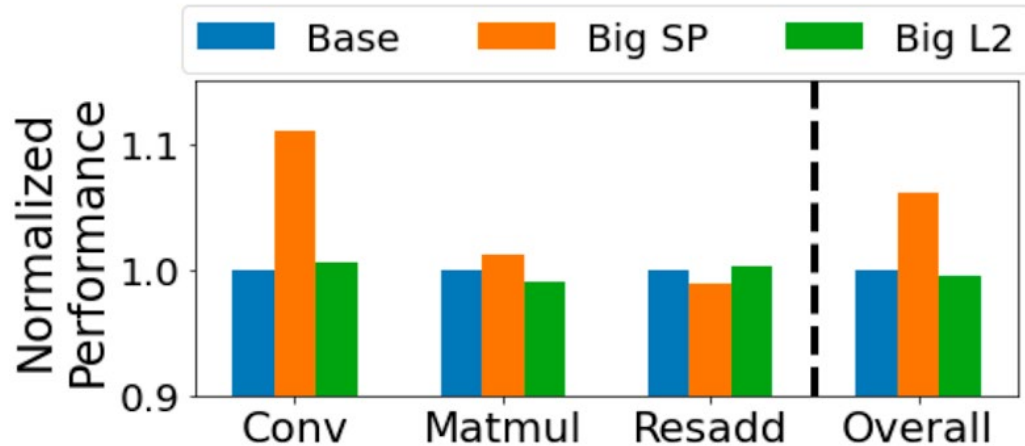
- Single core
 - Private scratchpad more helpful
 - Much better for convs





Case Study: Memory Partitioning

- Single core
 - Private scratchpad more helpful
 - Much better for convs
- Dual core
 - Shared L2 more helpful
 - Much better for residual additions





Conclusion

- Gemmini is:
 - Full-system
 - Full-stack
- Enables DSE and hardware/software co-design
 - Layer composition vs. memory partitioning
 - Virtual address translation design
- Open-source!
 - github.com/ucb-bar/gemmini

