ETH zürich

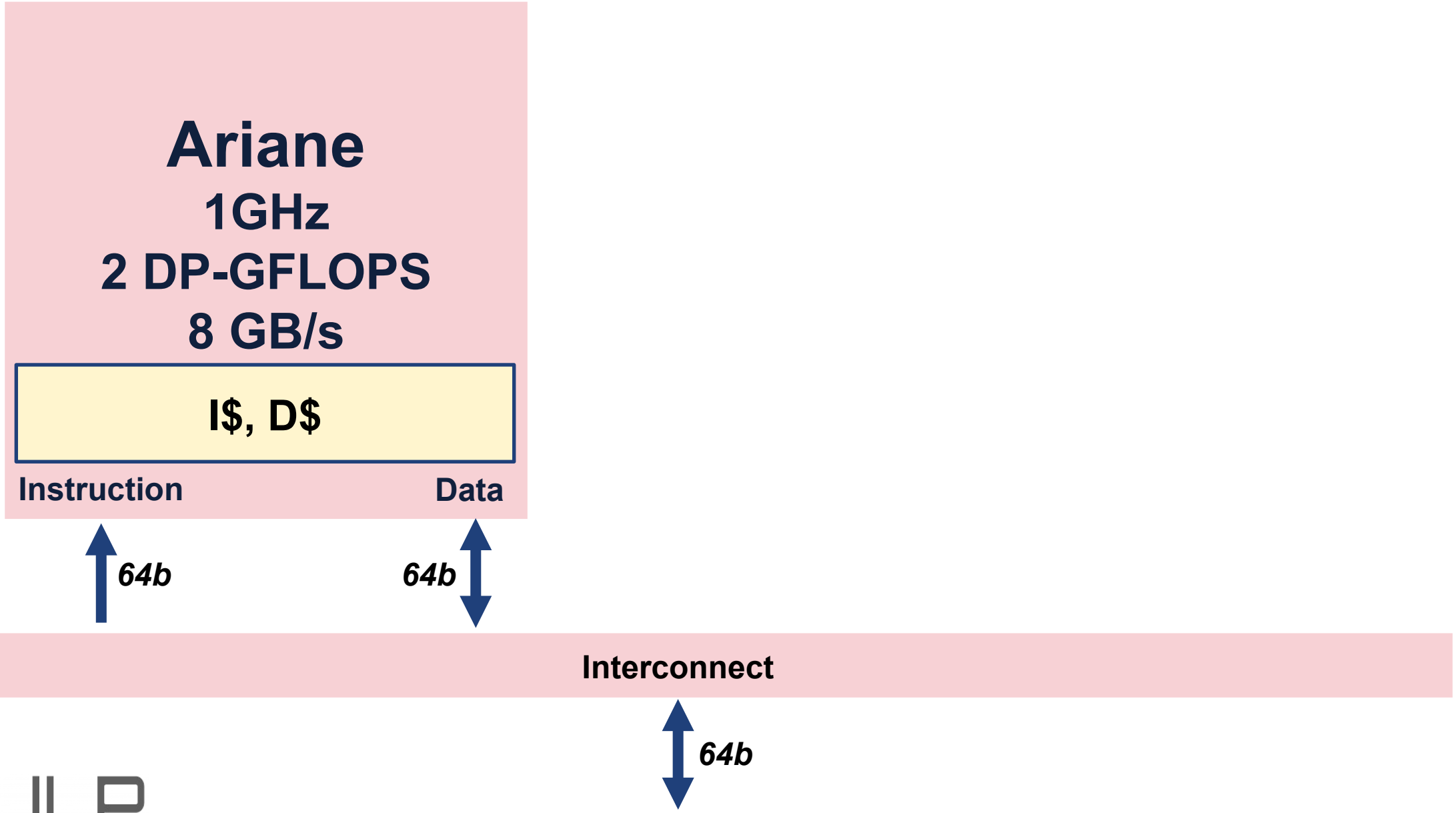
Ara

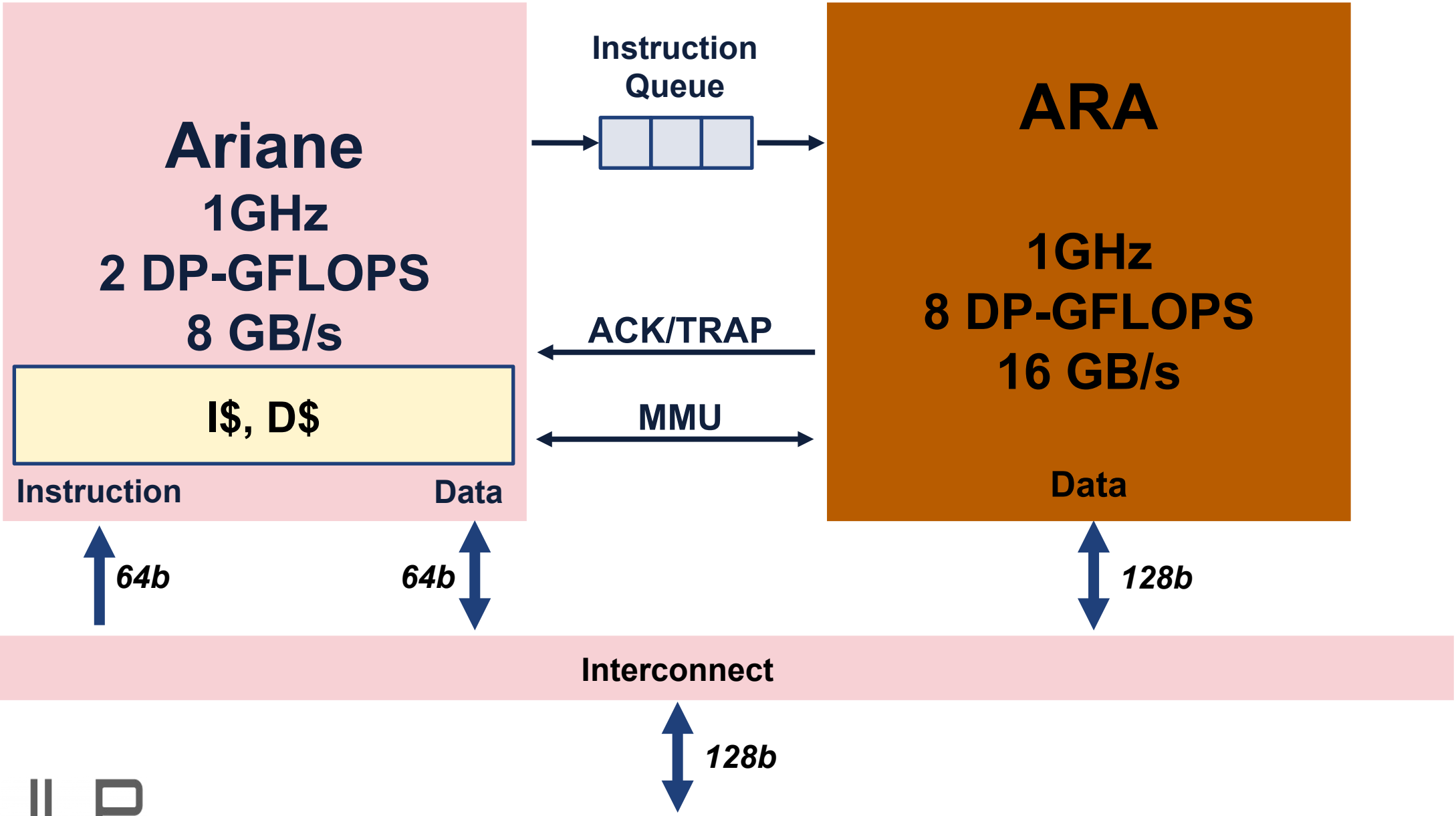
Design and implementation of a 1GHz+ 64-bit RISC-V Vector Processor in 22 nm FD-SOI

Matheus CAVALCANTE
PhD Student – ETH Zurich

Fabian SCHUIKI, Florian ZARUBA, Michael SCHAFFNER, Luca BENINI

PULP
Parallel Ultra Low Power

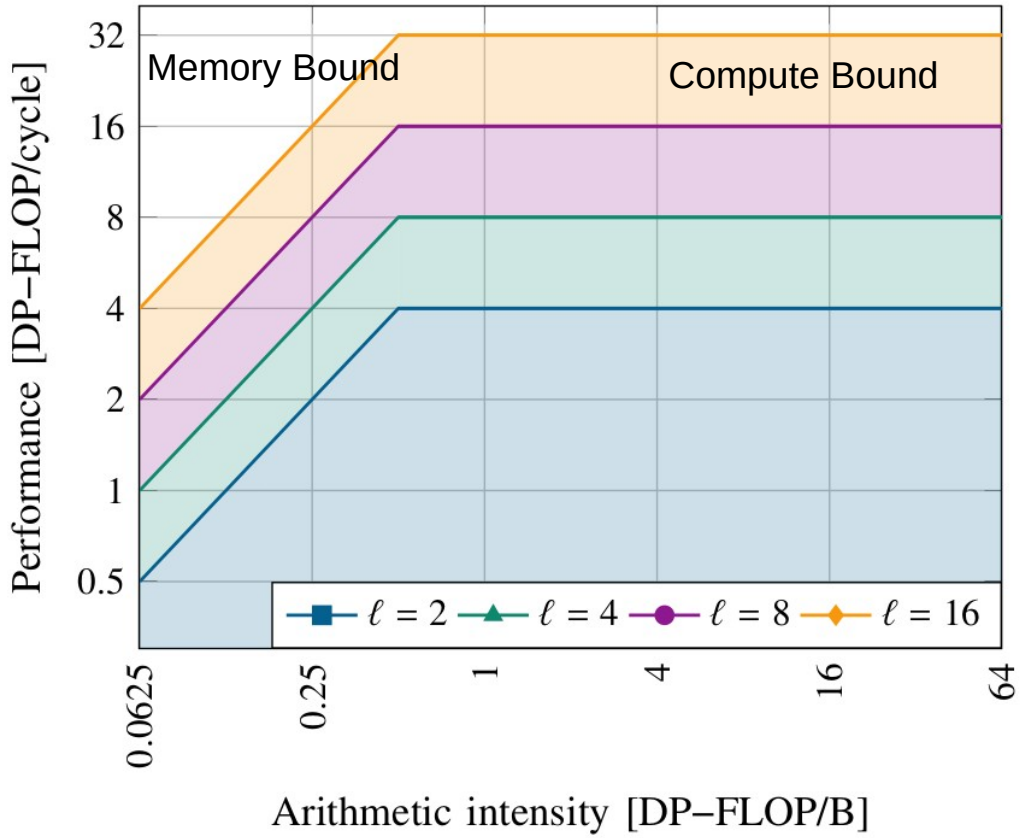

Ariane
1GHz
2 DP-GFLOPS
8 GB/s
I$, D$
Instruction
Data
64b
64b
Interconnect
64b



PULP
Parallel Ultra Low Power

Memory Bandwidth and Performance: Rooflines

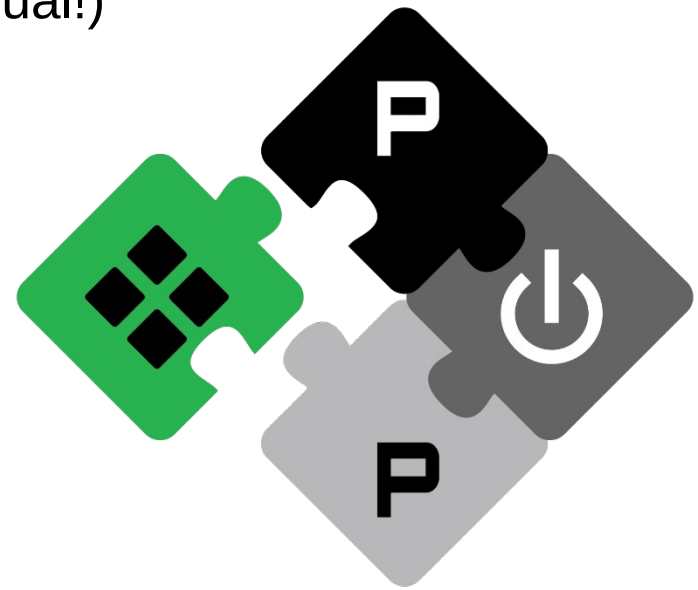
- Arithmetic Intensity
  - *Operations per byte*: data reuse of an algorithm
  - One FMA is two operations
- Memory-bound and compute-bound
  - Peak perf. per memory width ratio
- Ara targets 0.5 DP-FLOP/B
  - Memory bandwidth scales with the number of FMAs



PULP Parallel Ultra Low Power

Ara: High-performance vector processor

- GlobalFoundries' GF22 FD-SOI process
  - Work initiated at my Master's Thesis
  - First presented at the 1st RISC-V Summit, last year
  - Will be open-sourced still in 2019 within the PULP Platform (as usual!)
- Snapshot of the current development
  - Challenges we faced
  - Results we achieved
  - Insights we gained



RISC-V Vector Extension

- RISC-V “V” Extension: “Cray-like” vector-SIMD approach
- Ara is based on version 0.5
  - Work being done to update it to the latest version of the spec (0.7)
  - Open-sourcing later this year

- Not fully-compliant
  - Limited support to fixed-point and vector atomics (not our focus)
  - Limited support for type promotions (e.g., 8b + 8b ← 64b) – hardware cost

State-of-the-art

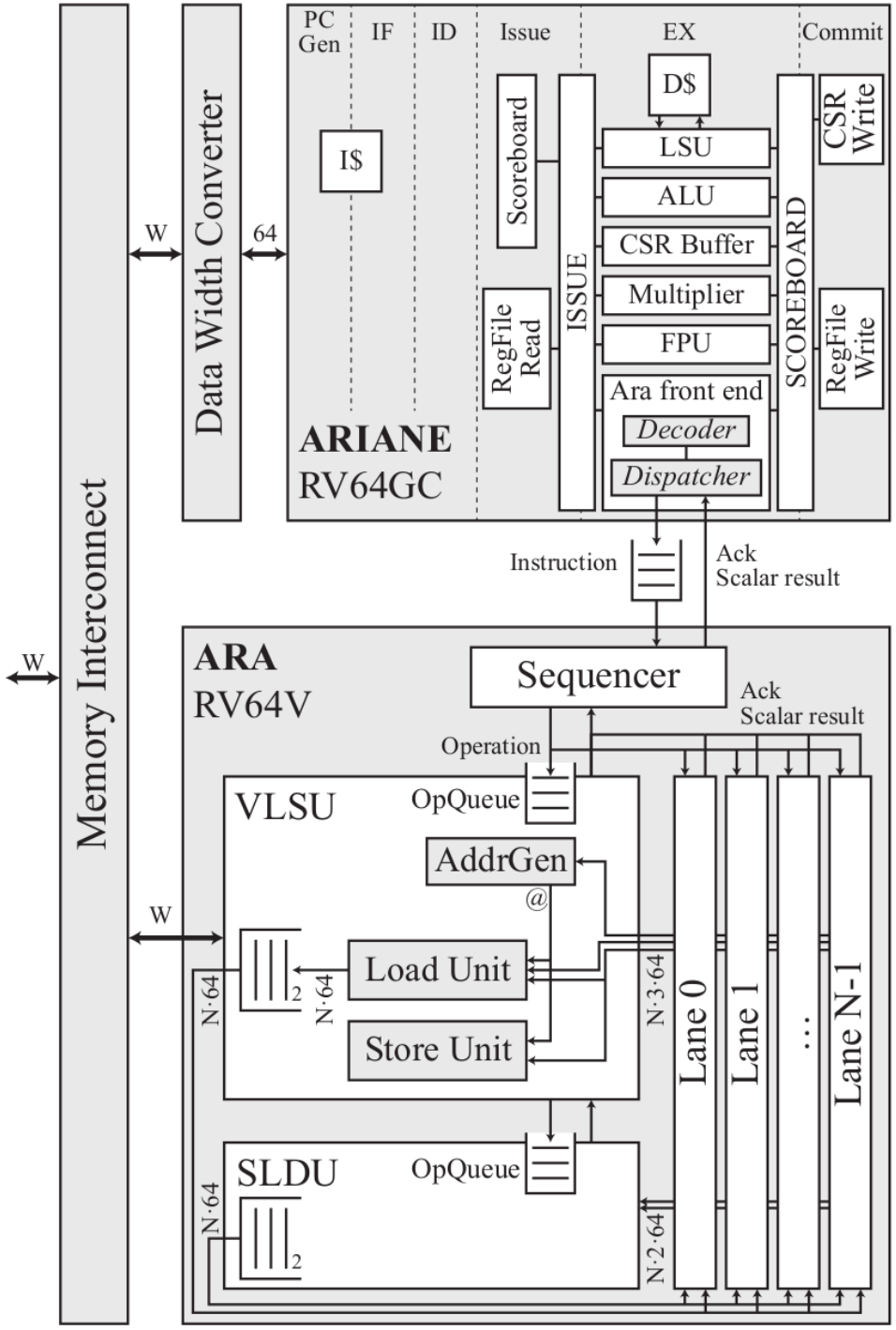
- Fujitsu's A64FX
  - Based on ARM SVE
  - 2.7 DP-TFLOPS at a 7 nm process

- Hwacha
  - Vector-fetch architecture
  - More complex: vector unit fetches its own instructions and threads can diverge
  - Predecessor to RISC-V "V" with its own ISA
    - Later version should be compliant with the vector extension
  - 64 DP-GFLOPS at TSMC 16 nm
  - 40 DP-GFLOPS/W at 28 nm process

ETH zürich

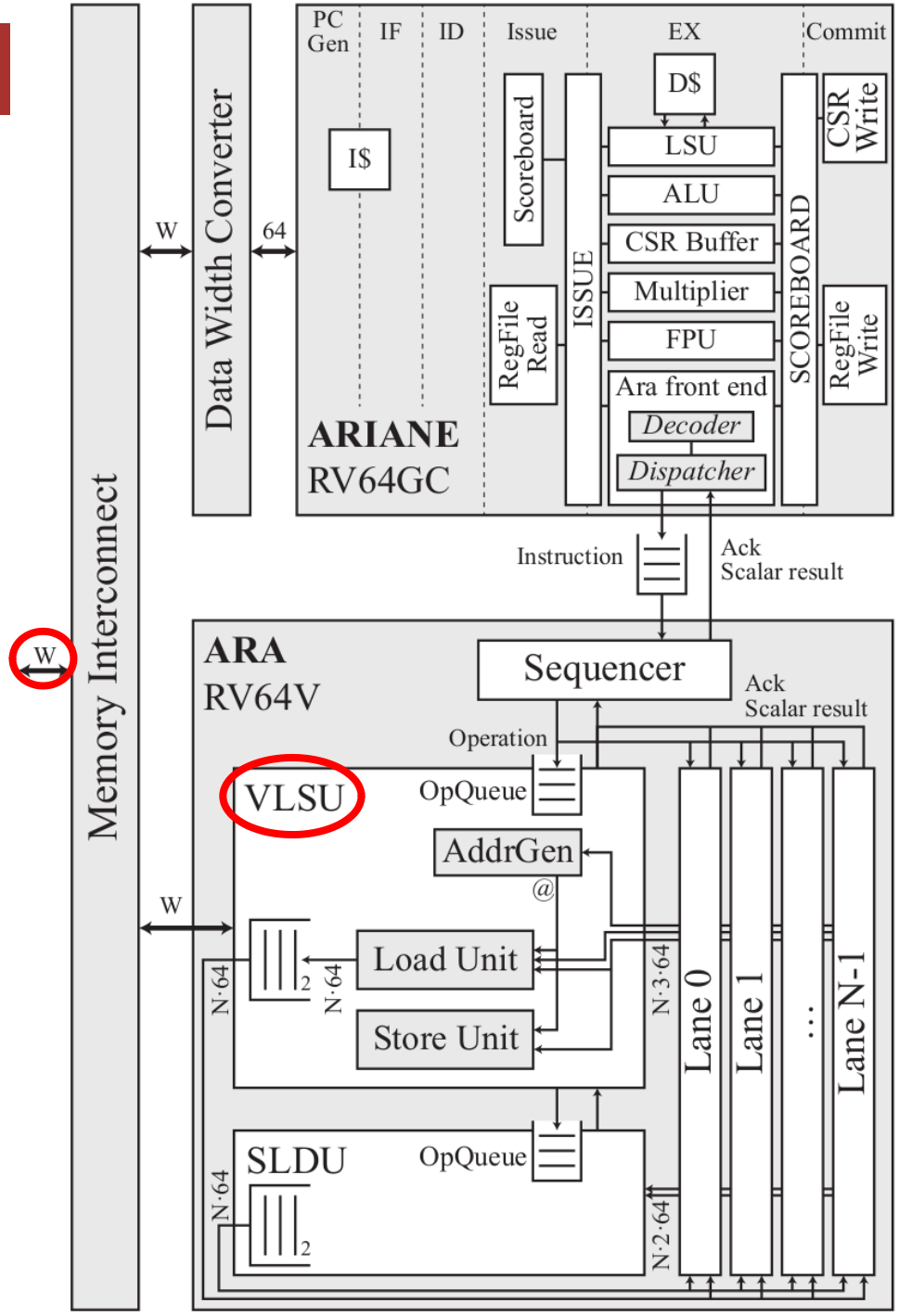
Microarchitecture

Ara with N identical lanes



Ara with N identical lanes

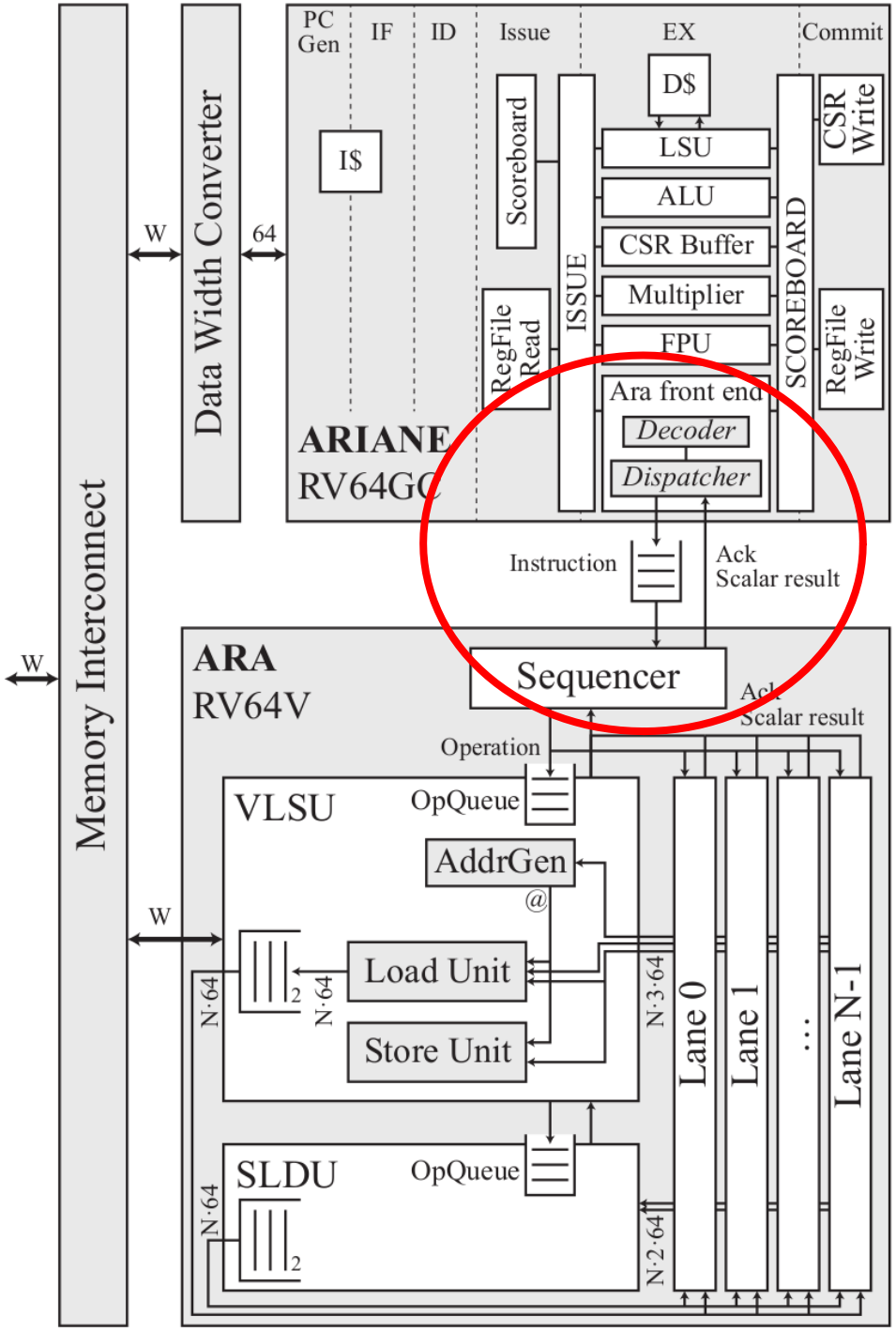
- Memory width *W*
  - Keep the peak perf. per memory width at 0.5 DPFLOP/B



PULP
Parallel Ultra Low Power

Ara with N identical lanes

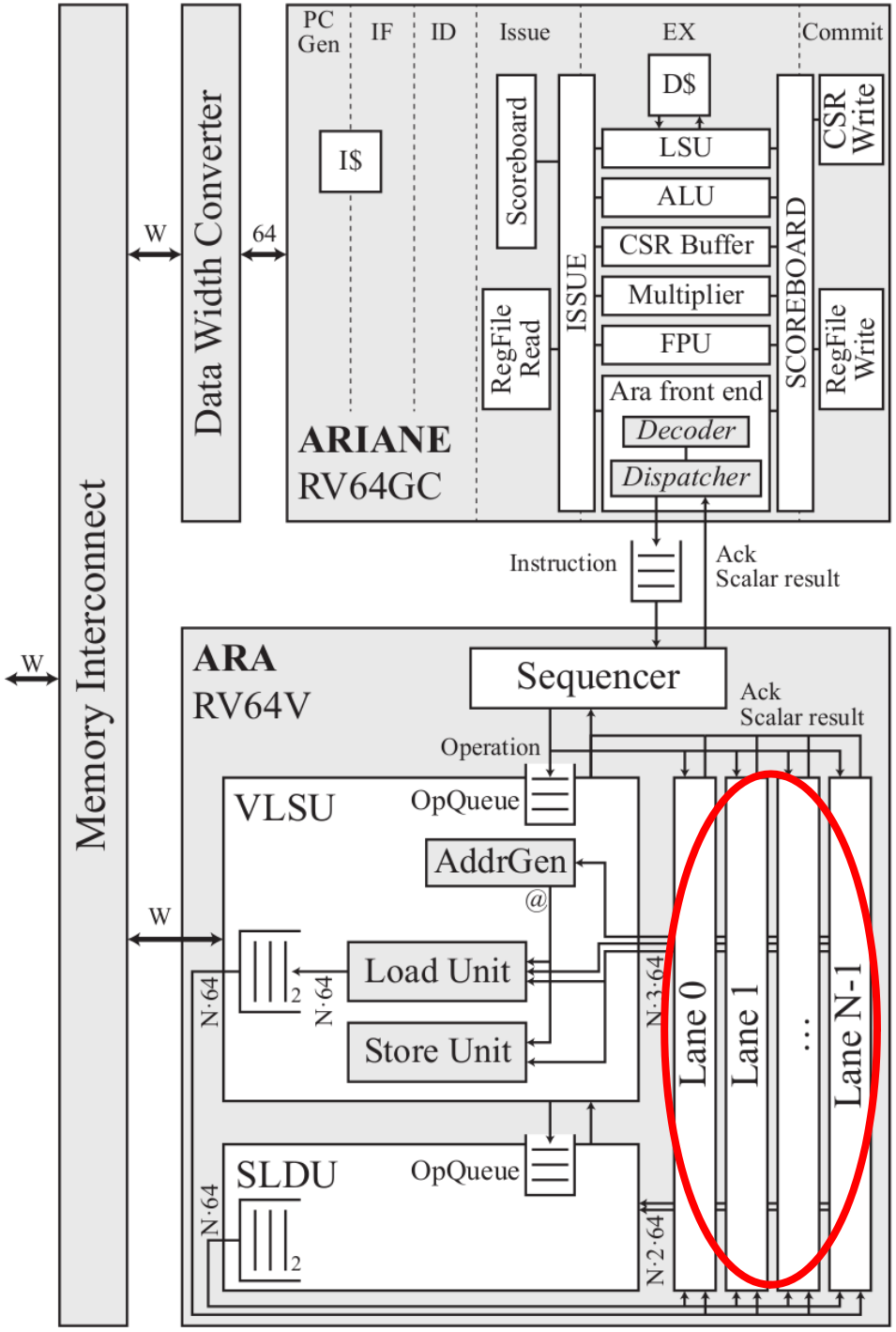
- Memory width *W*
  - Keep the peak perf. per memory width at 0.5 DPFLOP/B
- Vector instruction dispatching
  - Ara executes instructions *non-speculatively*
  - Sequencer acknowledges instructions as soon as they are deemed “safe”



PULP
Parallel Ultra Low Power

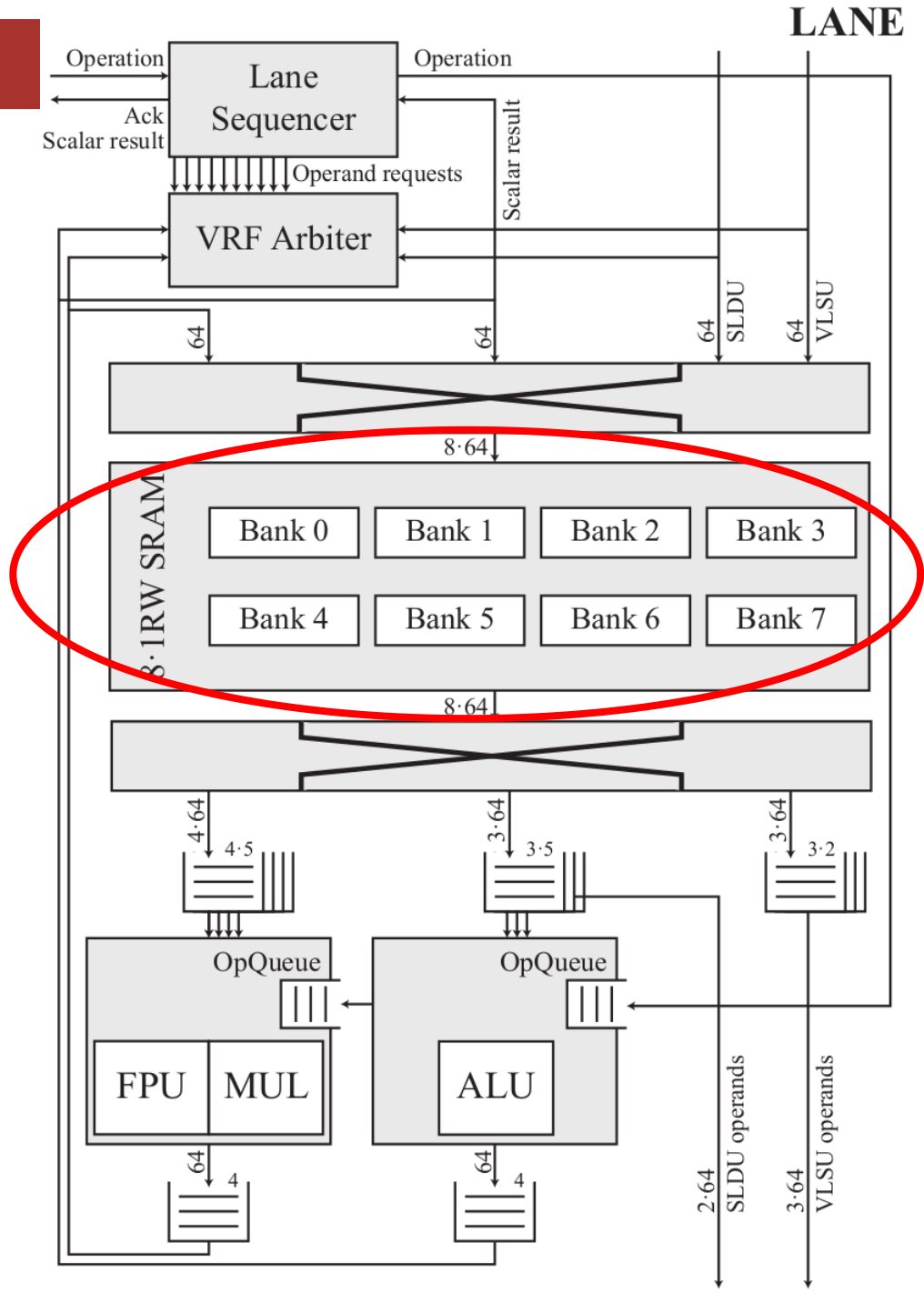
Ara with N identical lanes

- Memory width *W*
  - Keep the peak perf. per memory width at 0.5 DPFLOP/B
- Vector instruction dispatching
  - Ara executes instructions *non-speculatively*
  - Sequencer acknowledges instructions as soon as they are deemed “safe”
- Identical lanes
  - Each lane holds part of the computing units and part of the Vector Register File (VRF): scalability!



Lane microarchitecture

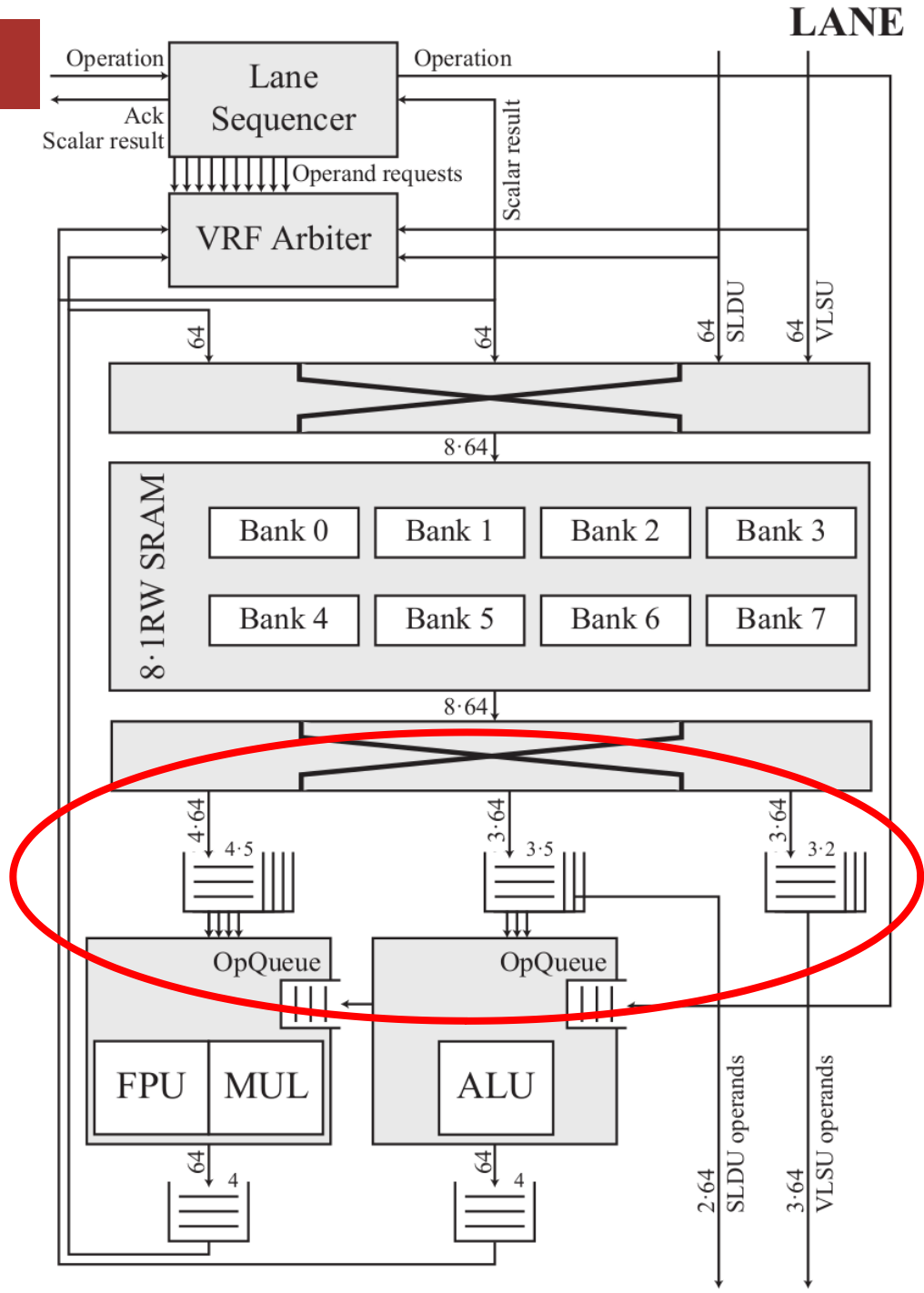
- Multibanked Vector Register File
  - Sustains high throughput without multiple ports
  - Requires an VRF Arbiter (banking conflicts)
  - Word width: 64 bits (aka operand width)



Lane microarchitecture

- Multibanked Vector Register File
  - Sustains high throughput without multiple ports
  - Requires an VRF Arbiter (banking conflicts)
  - Word width: 64 bits (aka operand width)

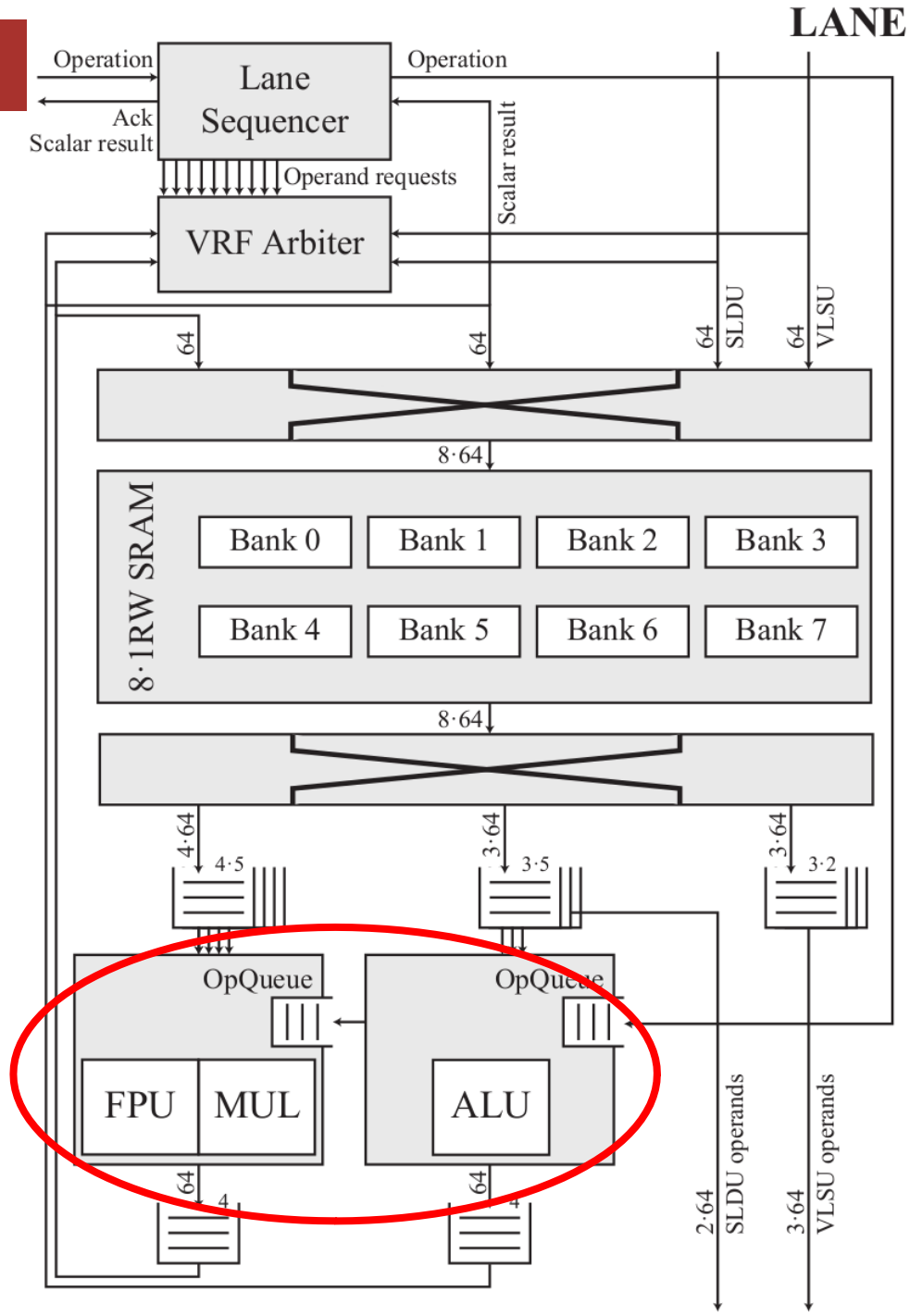
- Operand queues
  - Queues needed to sustain maximum throughput for the lock-step operation of the FUs, while hiding the latency caused by banking conflicts in the VRF



Trans-precision funcional units

- FPU can handle 1 x 64b, 2 x 32b, 4 x 16b and 8 x 8b per cycle
  - FMA is pipelined (5 cycles) to meet the fmax constraint
  - Design by Stefan Mach et al.

- Idea embedded in the ISA
  - CSR holds the “standard element width” of the vectors

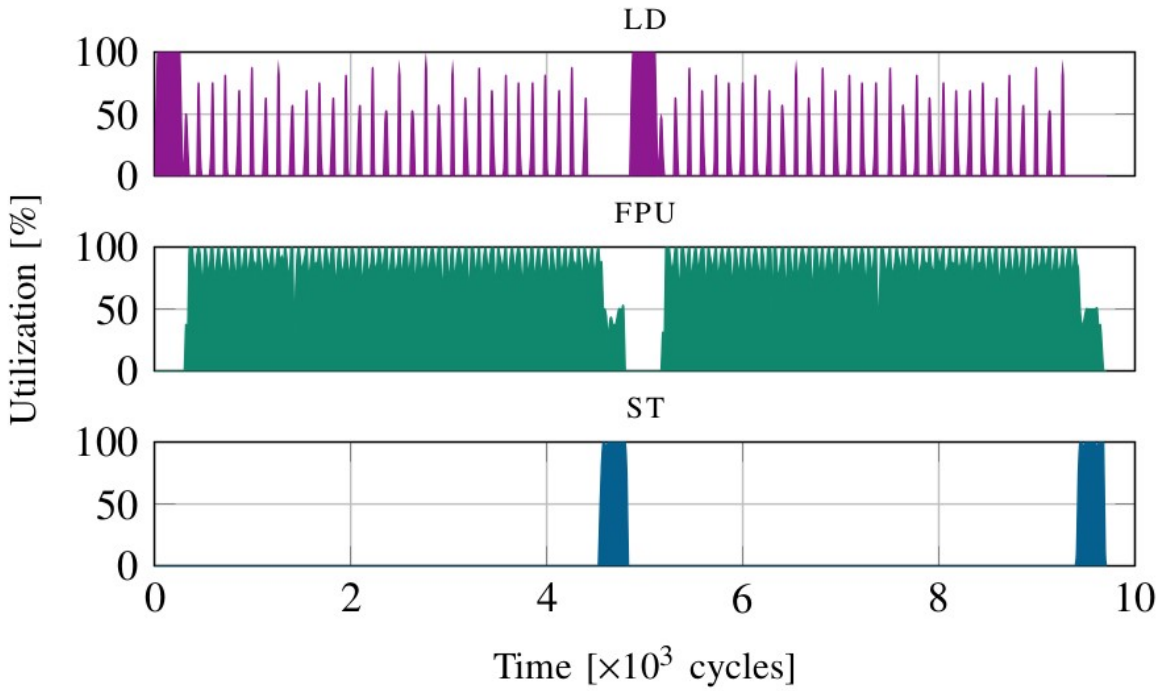


PULP
Parallel Ultra Low Power

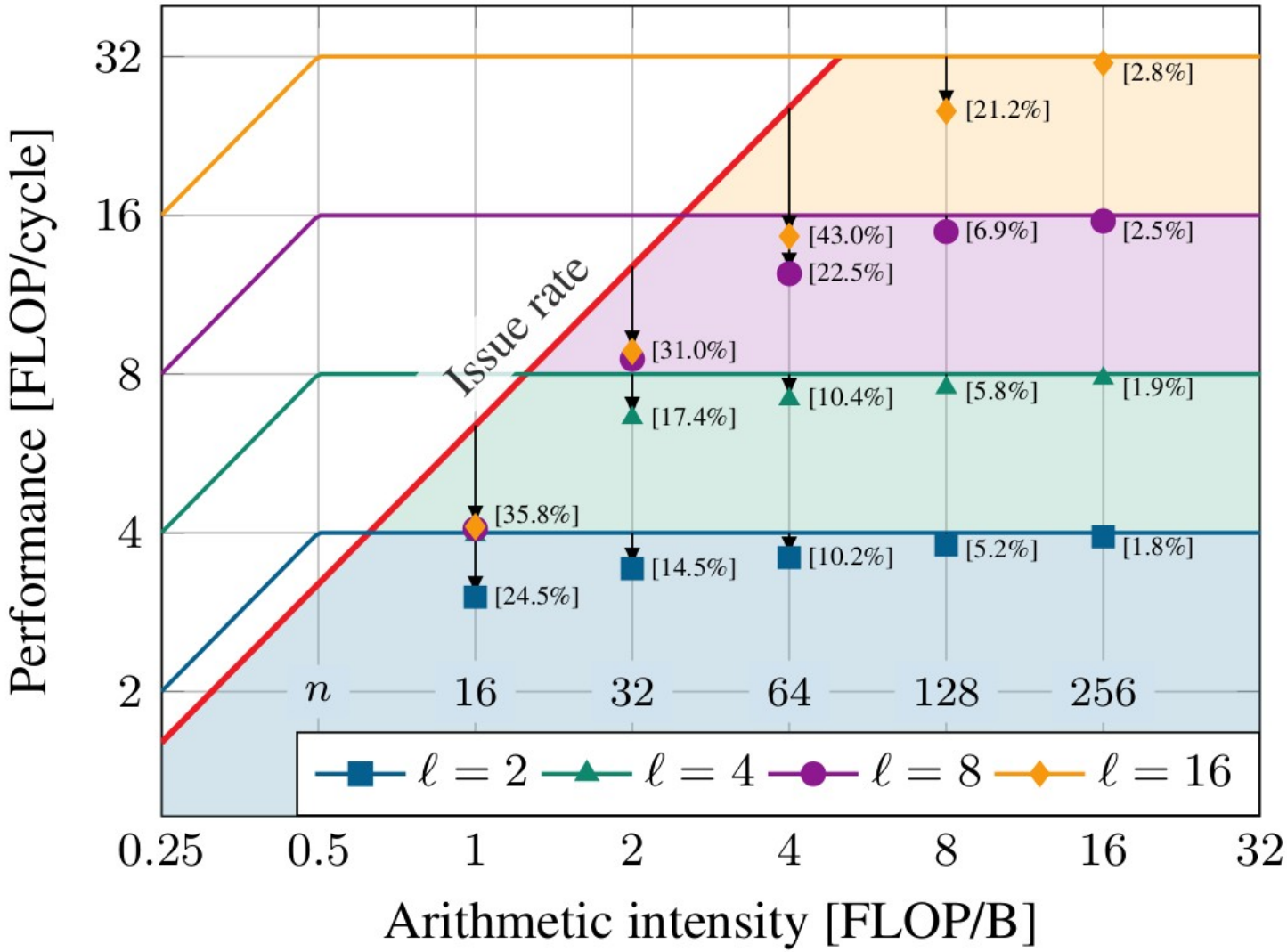
Performance Evaluation

Main kernel under evaluation: MATMUL

- DP-MATMUL: n x n double-precision matrix multiplication C ← AB + C
- $32n^2$ bytes of memory transfers and $2n^3$ operations
  - n/16 FLOP/B
  - Compute-bound on Ara for $n > 8$



Up to 98% efficiency @MATMUL (always?)



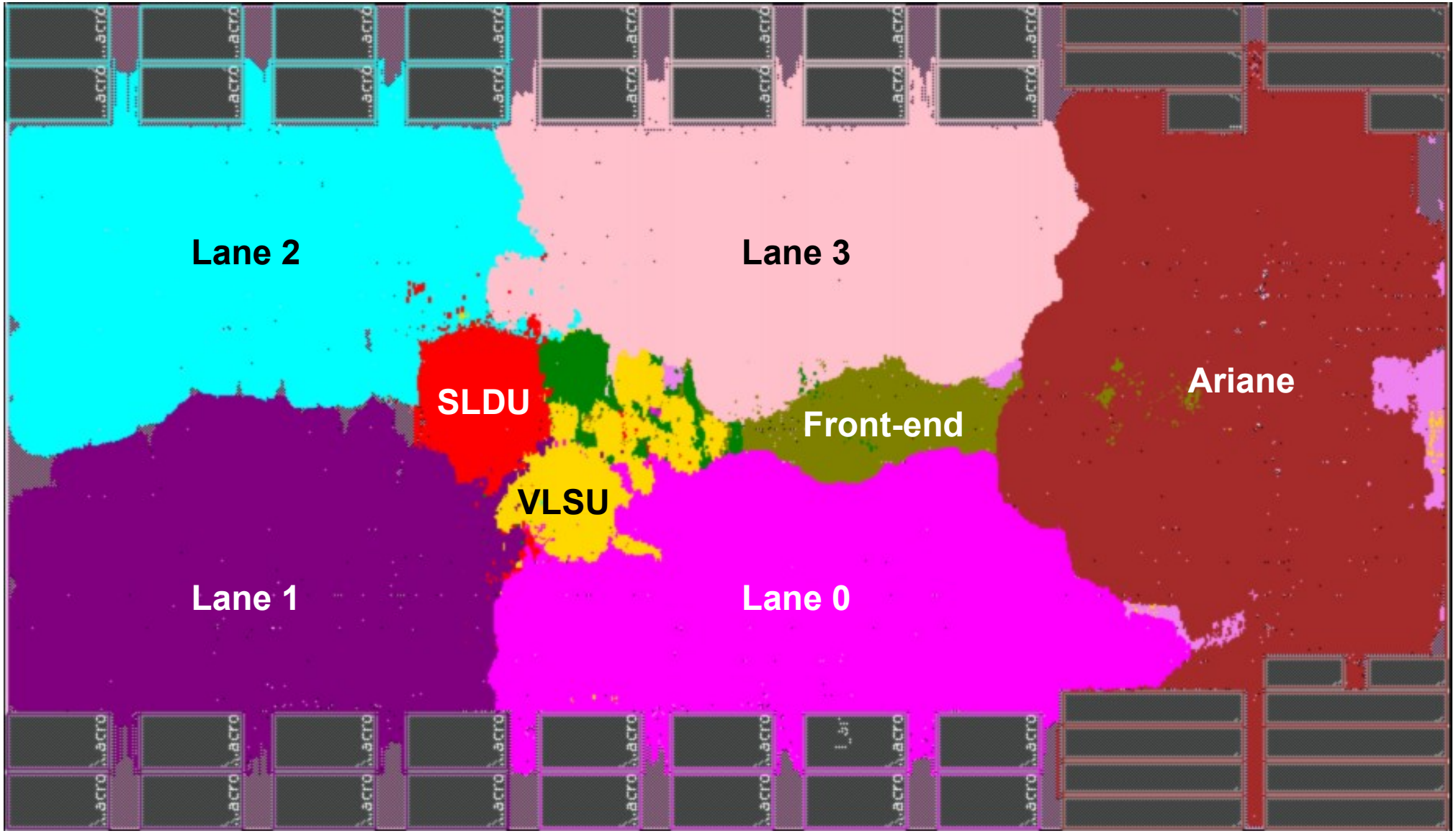
PULP
Parallel Ultra Low Power

Efficiency drop to 49% for a 16x16 MATMUL

vld vB, 0(a0)
ld t0, 0(a1)
add a1, a1, a2
vins vA, t0, zero
vmadd vC0, vA, vB, vC0
ld t0, 0(a1)
add a1, a1, a2
vins vA, t0, zero
vmadd vC1, vA, vB, vC1
ld t0, 0(a1)
add a1, a1, a2
vins vA, t0, zero
vmadd vC2, vA, vB, vC2
...

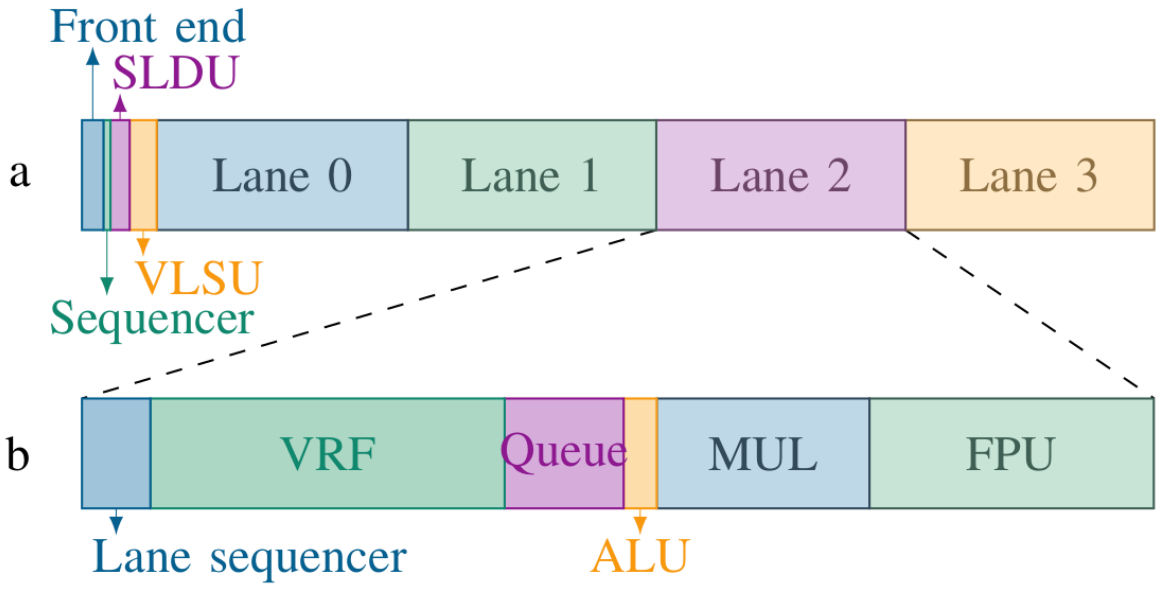
- *vmadd*s are issued at best every four cycles
  - Ariane is single-issue core
- If the *vmadd* takes less than four cycles to execute, the FPUs starve waiting for instructions
- This translates to the “issue rate” boundary on the roofline plot
  - Vector processor becomes more and more like an array processor

Ara: 4 lanes GF 22FDX 1.25 GHz implementation (TT, 0.80V, 25 ºC)



Figures of merit

- Area breakdown

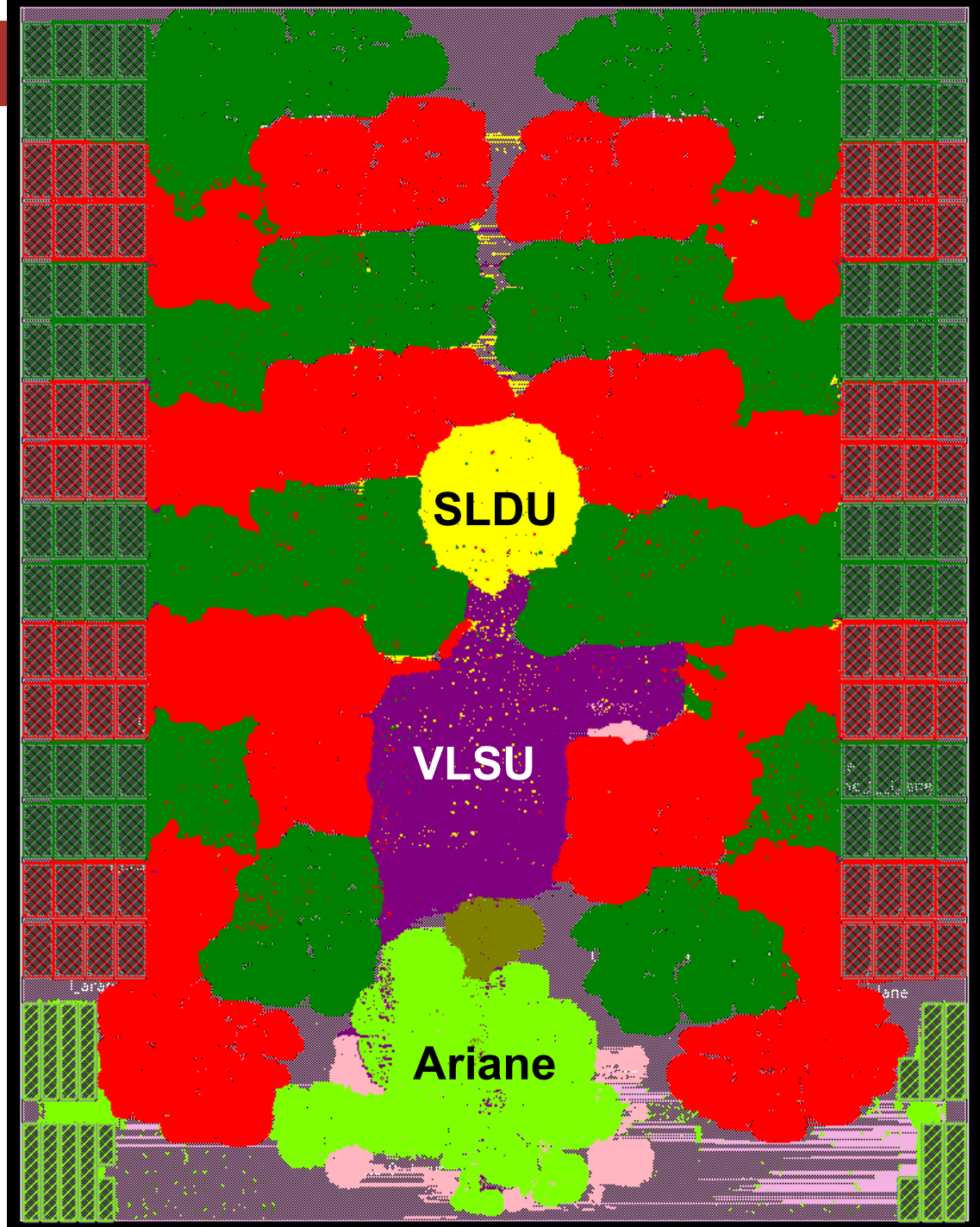


- Clock frequency:
  - 1.25 GHz (nominal), 0.92 GHz (worst)
- Area: 3430 kGE (0.68 mm^2)
- 256x256 MATMUL
  - Performance: 9.80 DP-GFLOPS
  - Power: 259 mW
  - Efficiency: 38 DP-GFLOPS/W

PULP
Parallel Ultra Low Power

Ara's scalability

- Each lane is *almost* independent
  - Contains part of the VRF and a FMA unit
- Scalability limitations
  - VLSU and SLDU: needs to communicate with all lanes, writing at all VRF banks
- Instance with 16 lanes achieves
  - 1.04 GHz (nominal), 0.78 GHz (worst)
  - 10.7 MGE (2.13mm^2)
  - 32.4 DP-GFLOPS
  - 40.8 DP-GFLOPS/W



More details?

- More details available in arXiv paper
  - *Ara: A 1 GHz+ Scalable and Energy-Efficient RISC-V Vector Processor with Multi-Precision Floating Point Support in 22 nm FD-SOI*
  - arxiv.org/abs/1906.00478
- Open-sourcing within PULP Platform
  - Planned for before the end of this year!
- Contact me at matheusd *at* iis.ee.ethz.ch :)

ETH zürich

Ara

Design and implementation of a 1GHz+ 64-bit RISC-V Vector Processor in 22 nm FD-SOI

Matheus CAVALCANTE
PhD Student – ETH Zurich

Fabian SCHUIKI, Florian ZARUBA, Michael SCHAFFNER, Luca BENINI

PULP
Parallel Ultra Low Power