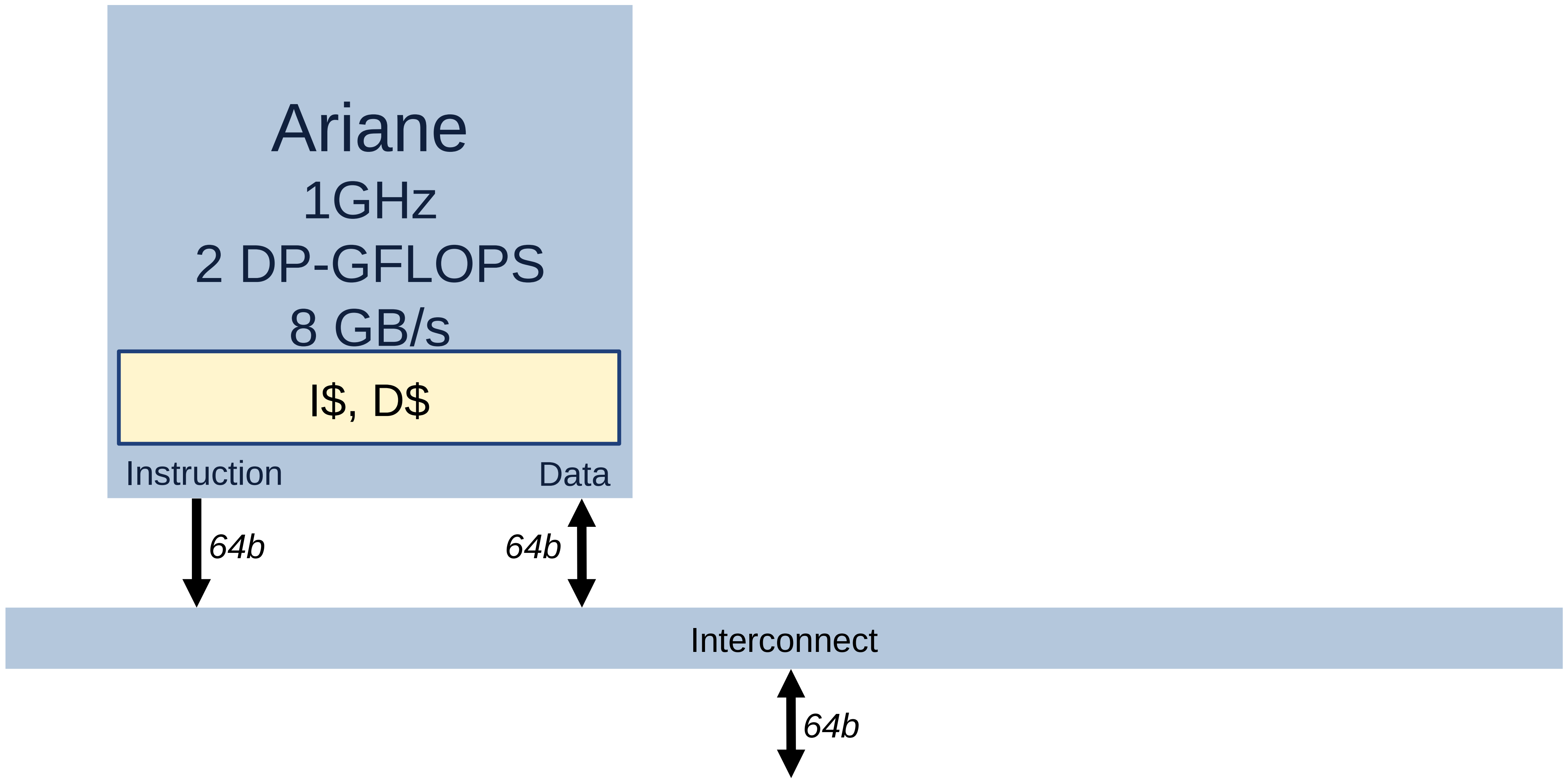
ETH zürich

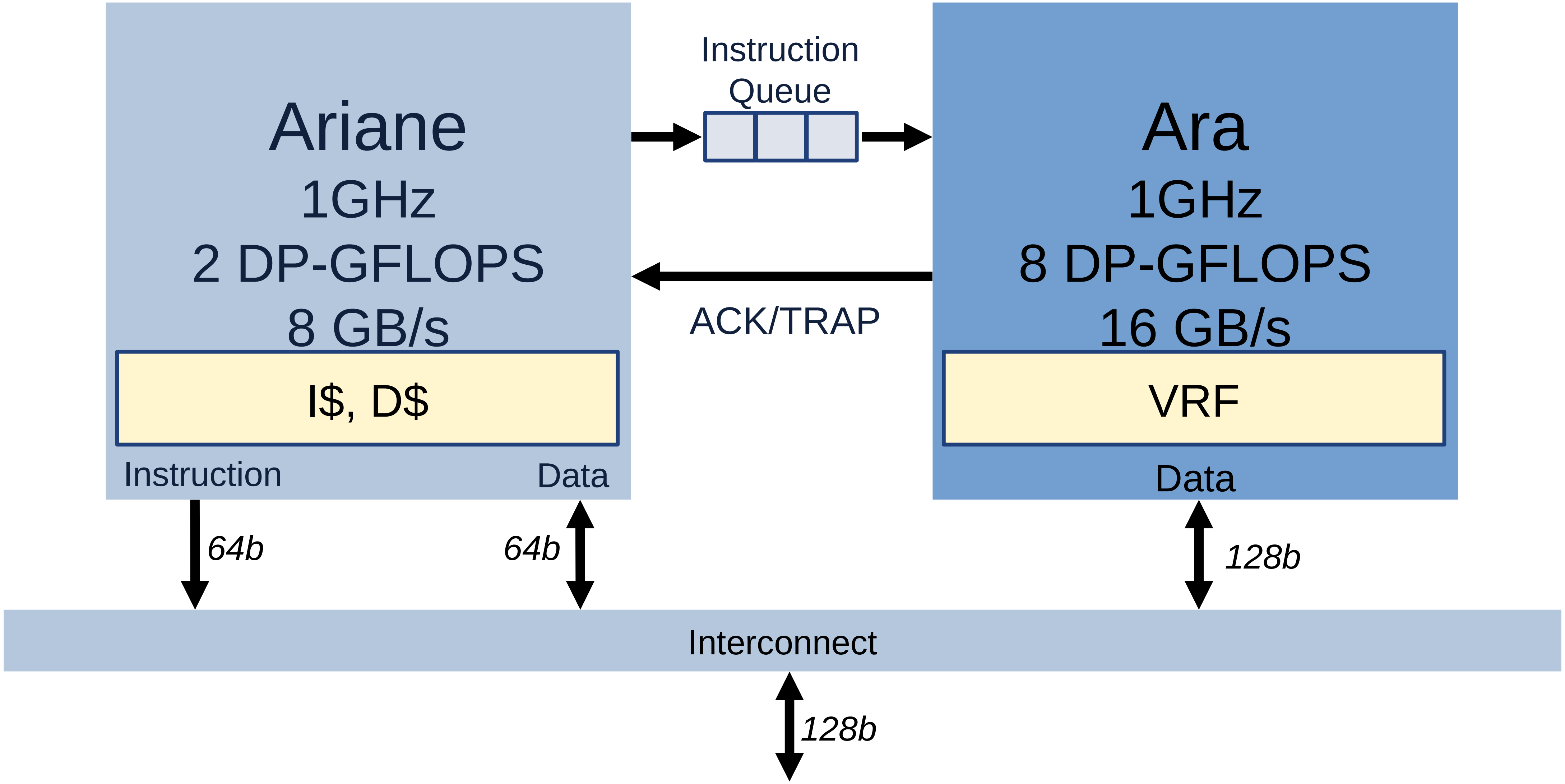
Ara: Status Update

Matteo Perotti
Matheus Cavalcante
PhD Students, Integrated Systems Laboratory
21 July 2021, Zurich

Ara: High-performance vector processor

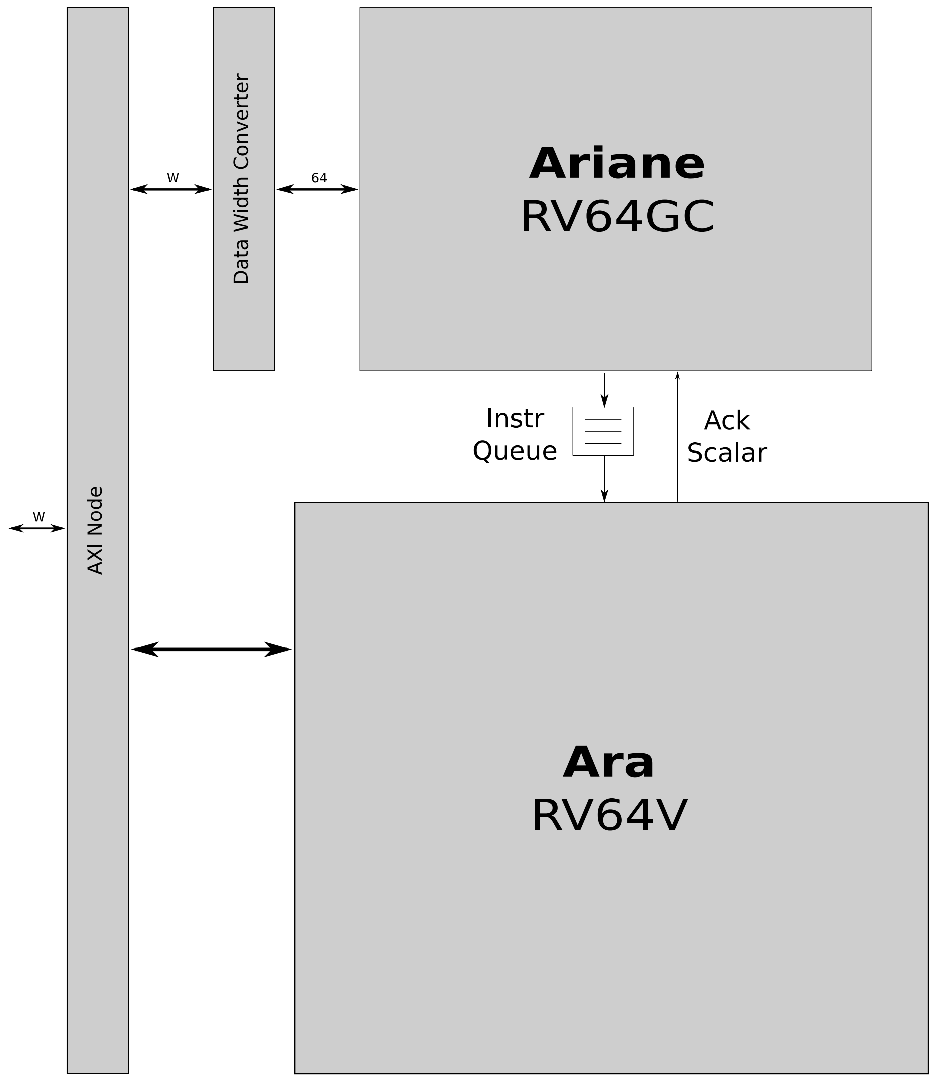


Ara: High-performance vector processor



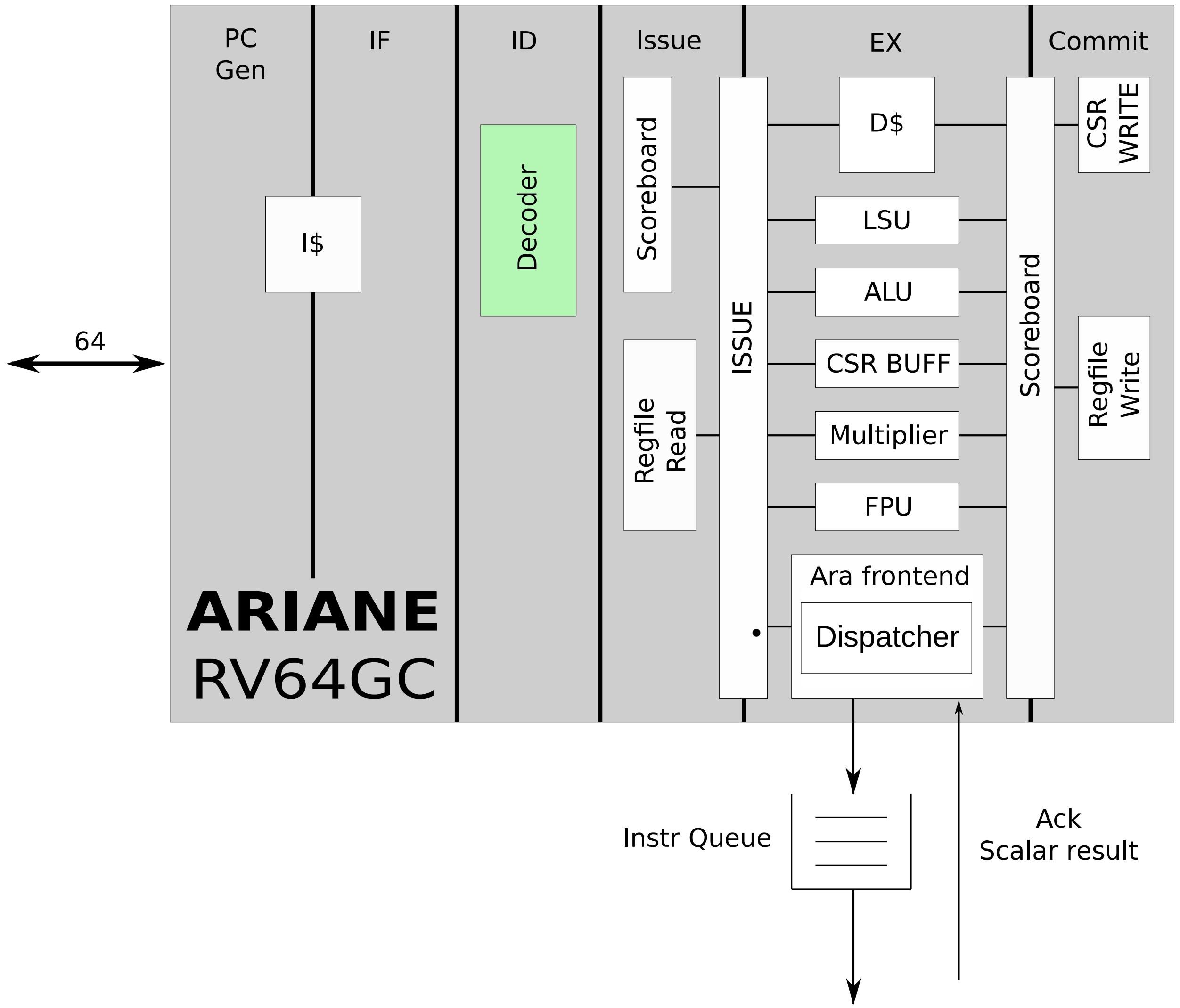
Ara - High level architecture

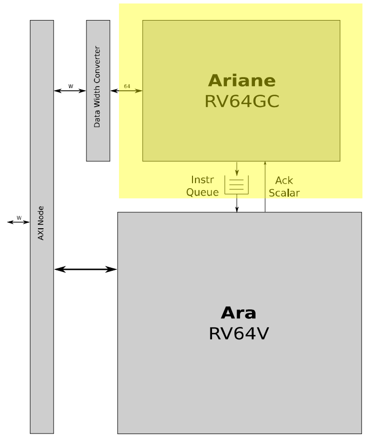
- **Coupled with Ariane**
- **Ariane dispatches RVV instructions to Ara**
- **Ara acknowledges Ariane** (Any errors? Scalar result?)
- Private memory access through AXI



Ariane – RVV instruction dispatch

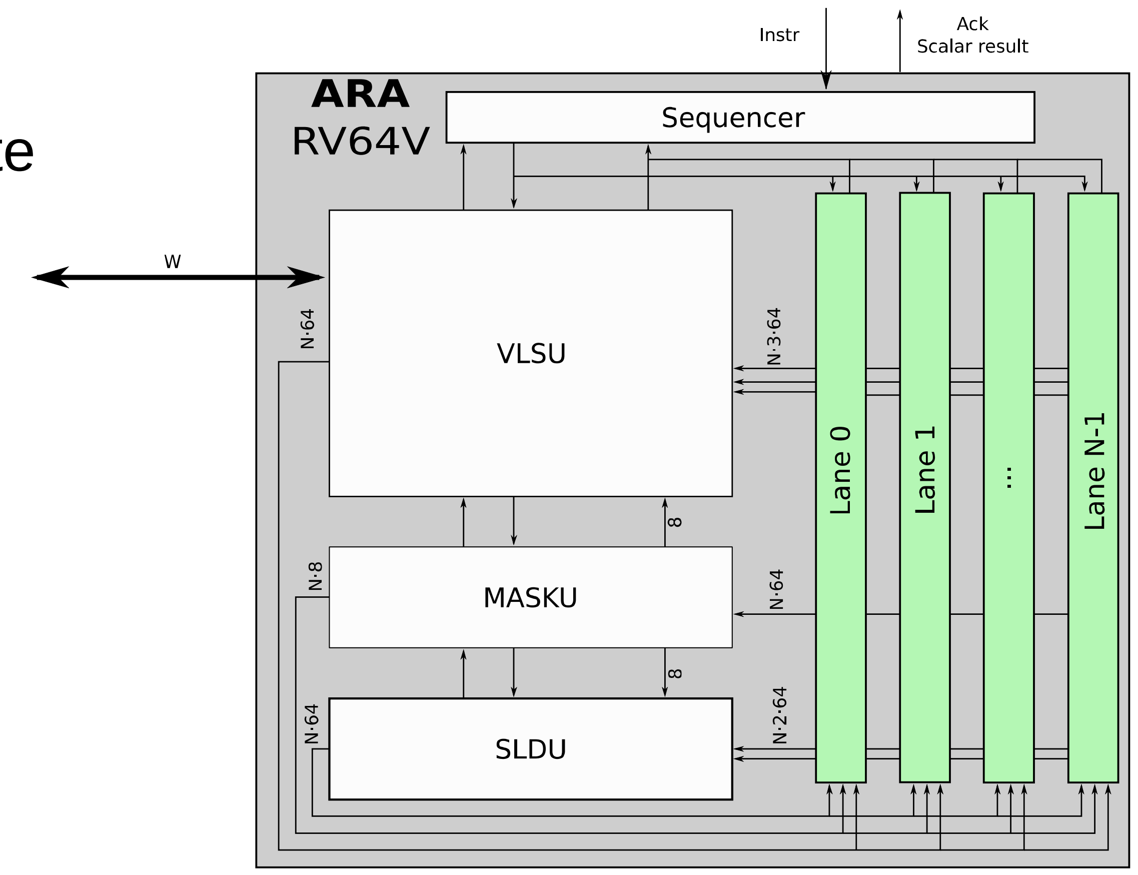
- Fetch + pre-decode in Ariane
- Scoreboard + **Issue to Ara frontend**
- Dispatch to Ara at **commit-time only**
- **Wait for Ara‘s answer to commit**

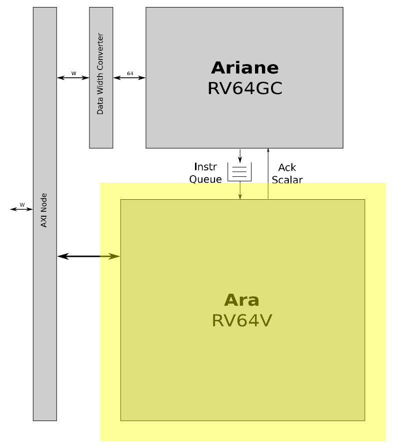




Ara – RVV instruction processing

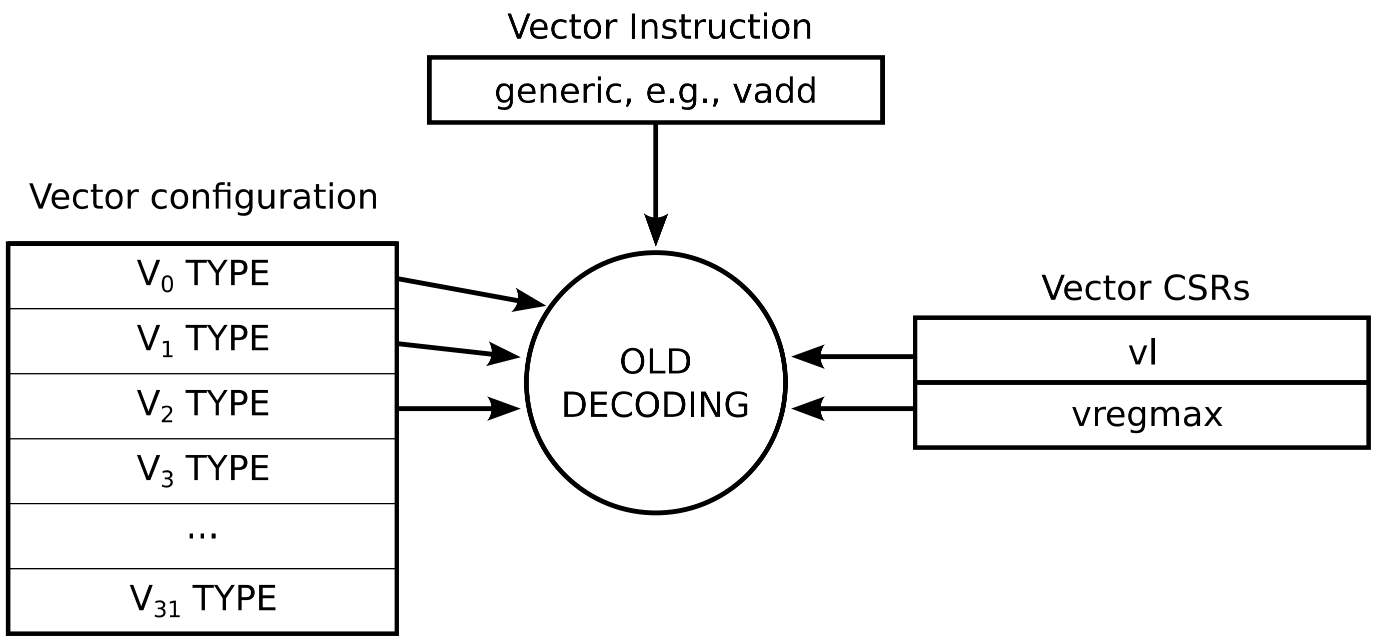
- **Sequencer**
  - Global view of Ara‘s state
- **VLSU**
  - **Memory** (AXI) ↔ **VRF** transfers
  - Address checks
- **VRF** split among the **lanes**
- **Lanes**: **VRF** + **processing**
- **New Units:**
  - **MASKU and SLDU**

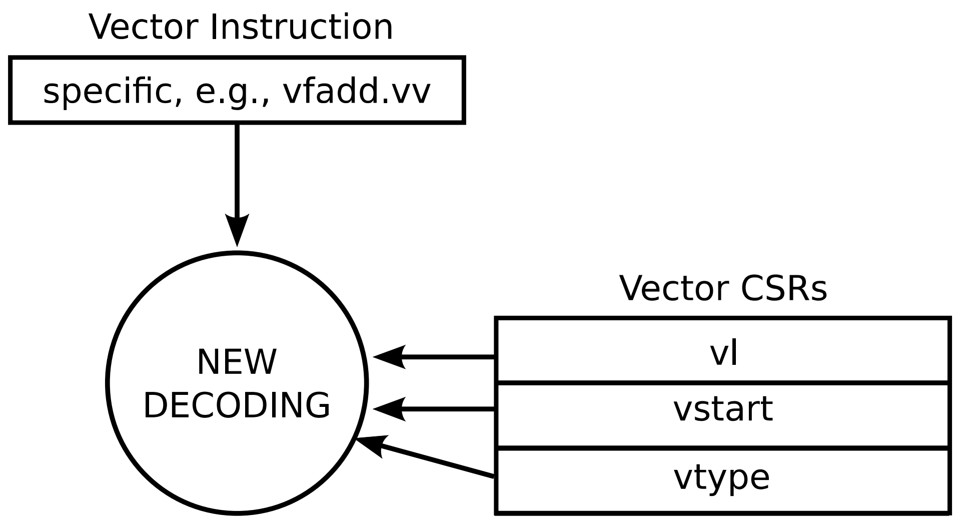




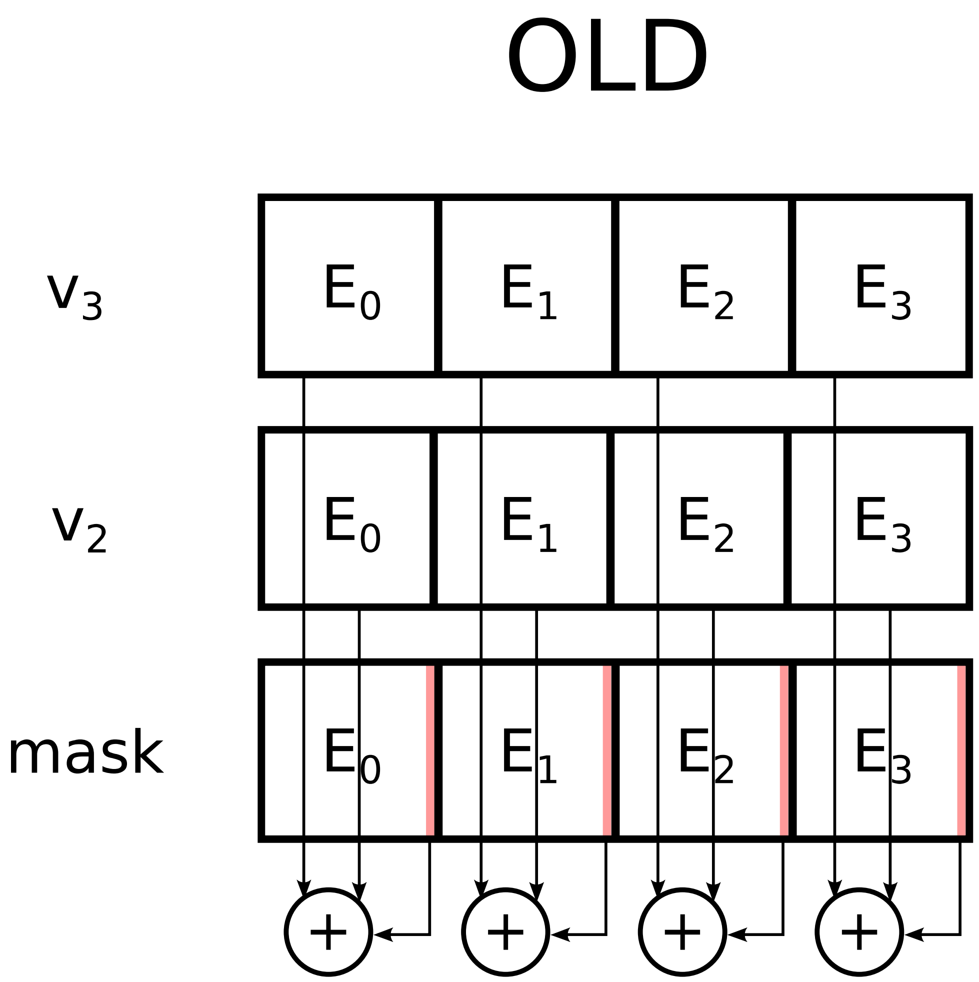
RVV 0.5 → 0.10 changes - Encoding

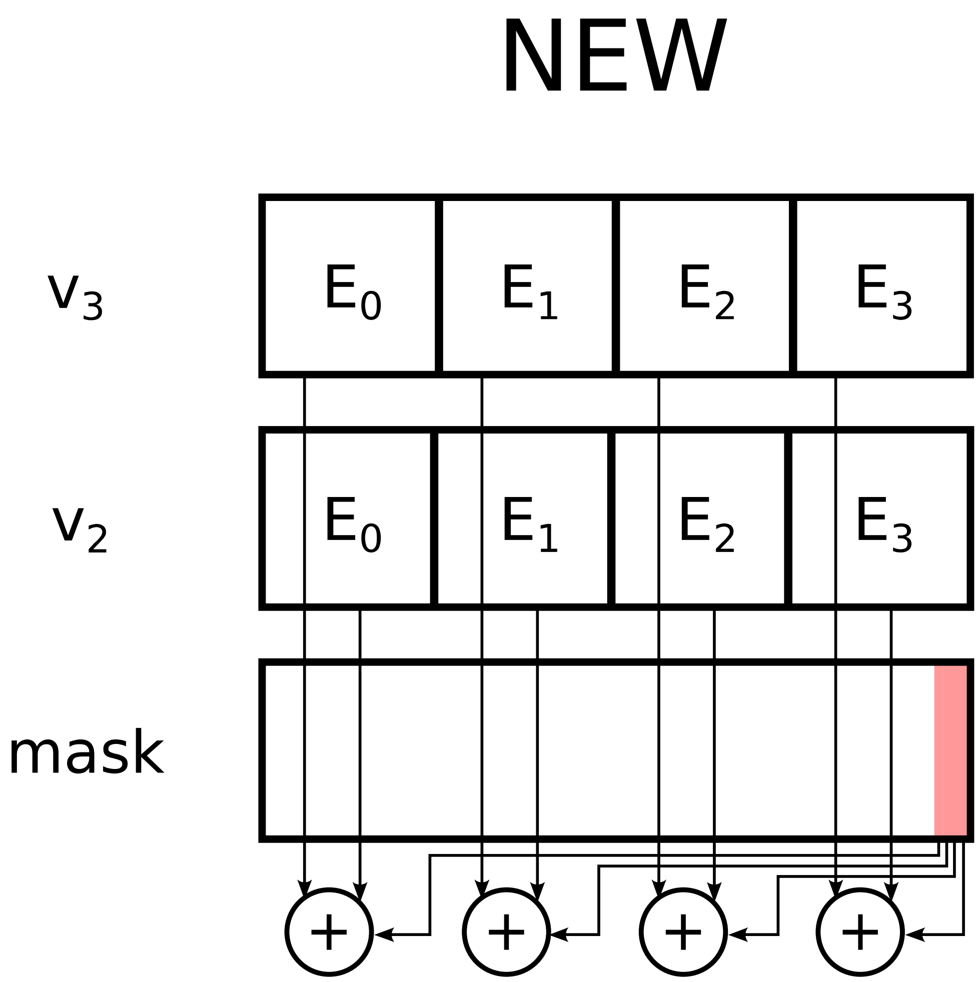
- **RVV 0.5 → Polimorphic encoding**
- **RVV 0.10 → Monomorphic encoding**



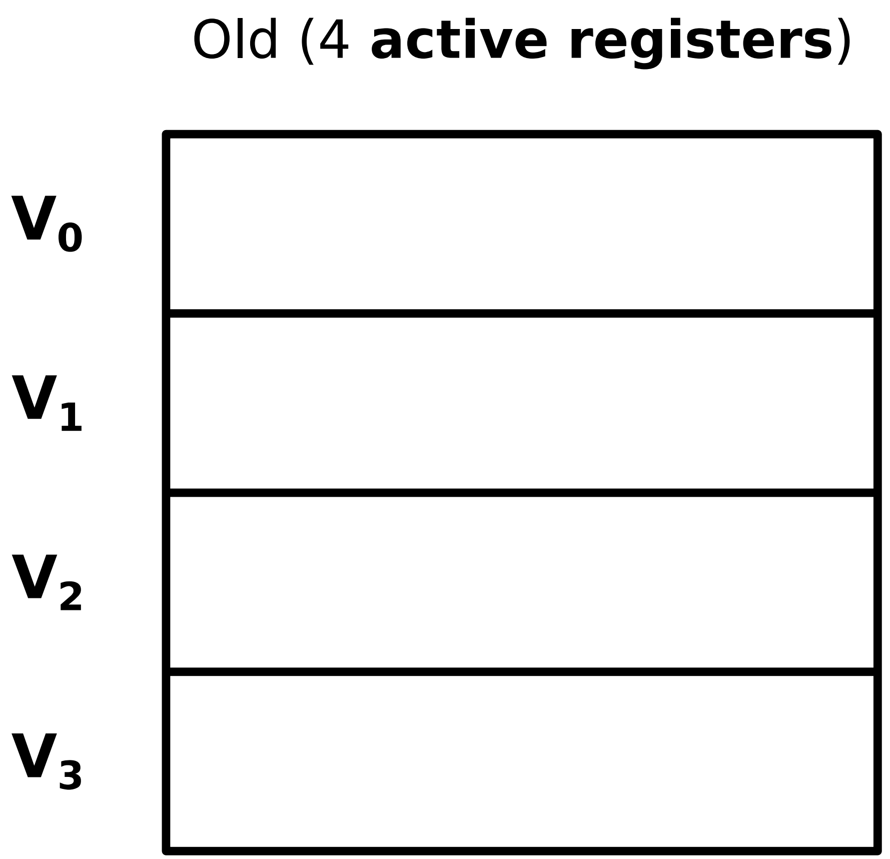


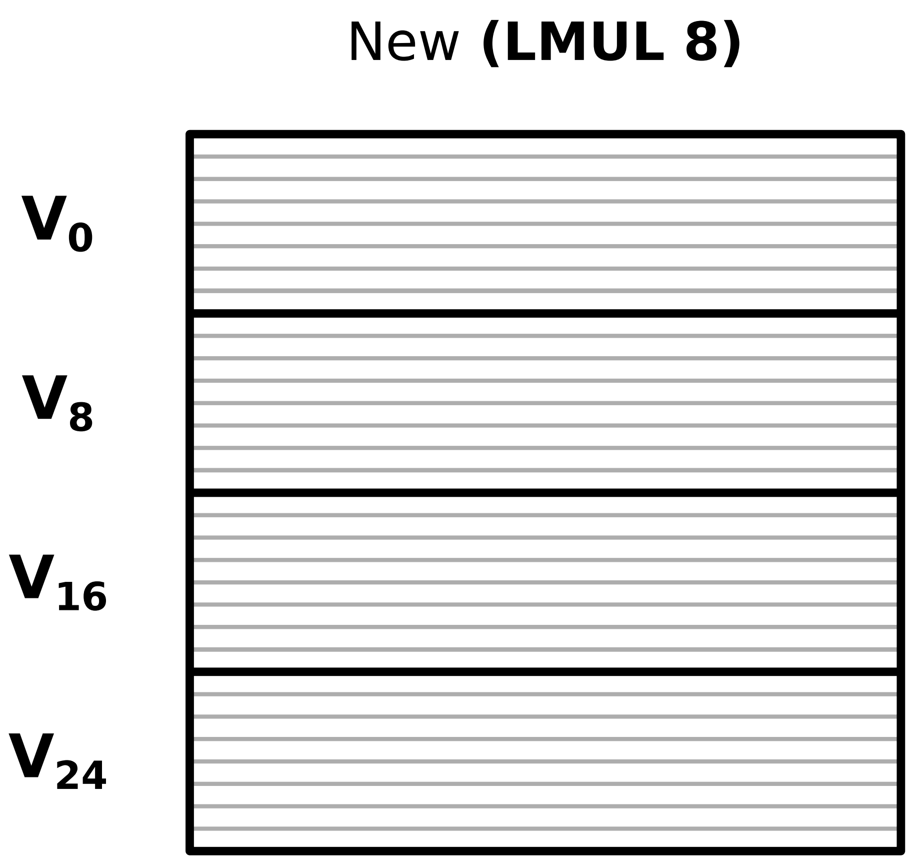
RVV 0.5 → 0.10 changes - Mask layout





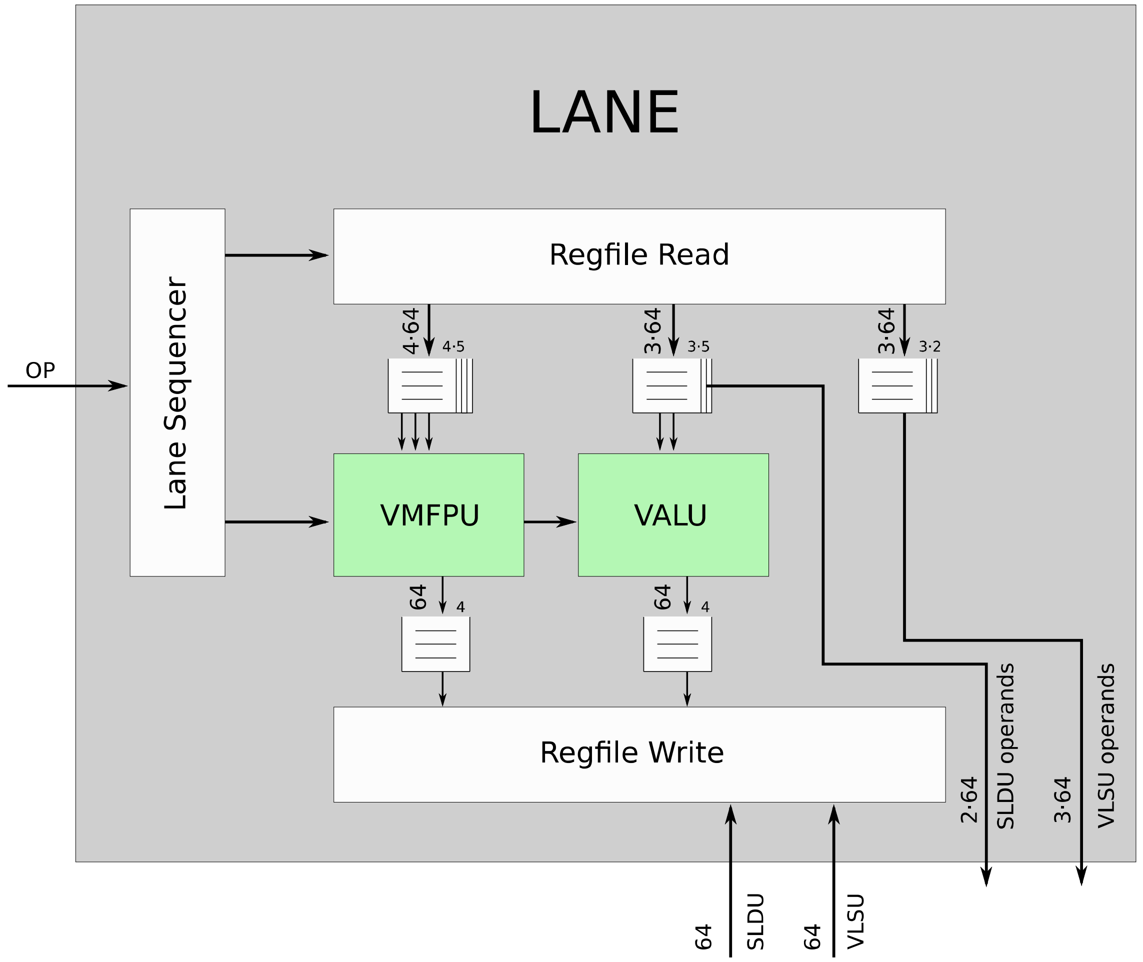
RVV 0.5 → 0.10 changes - Vector length modification

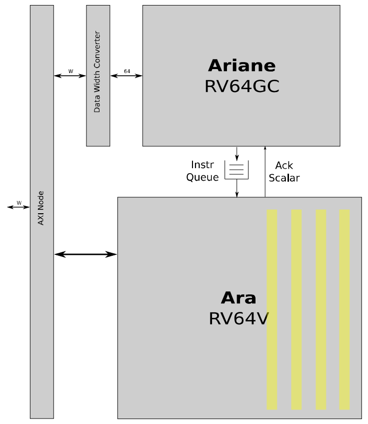




Ara Lane - RVV instruction core processing

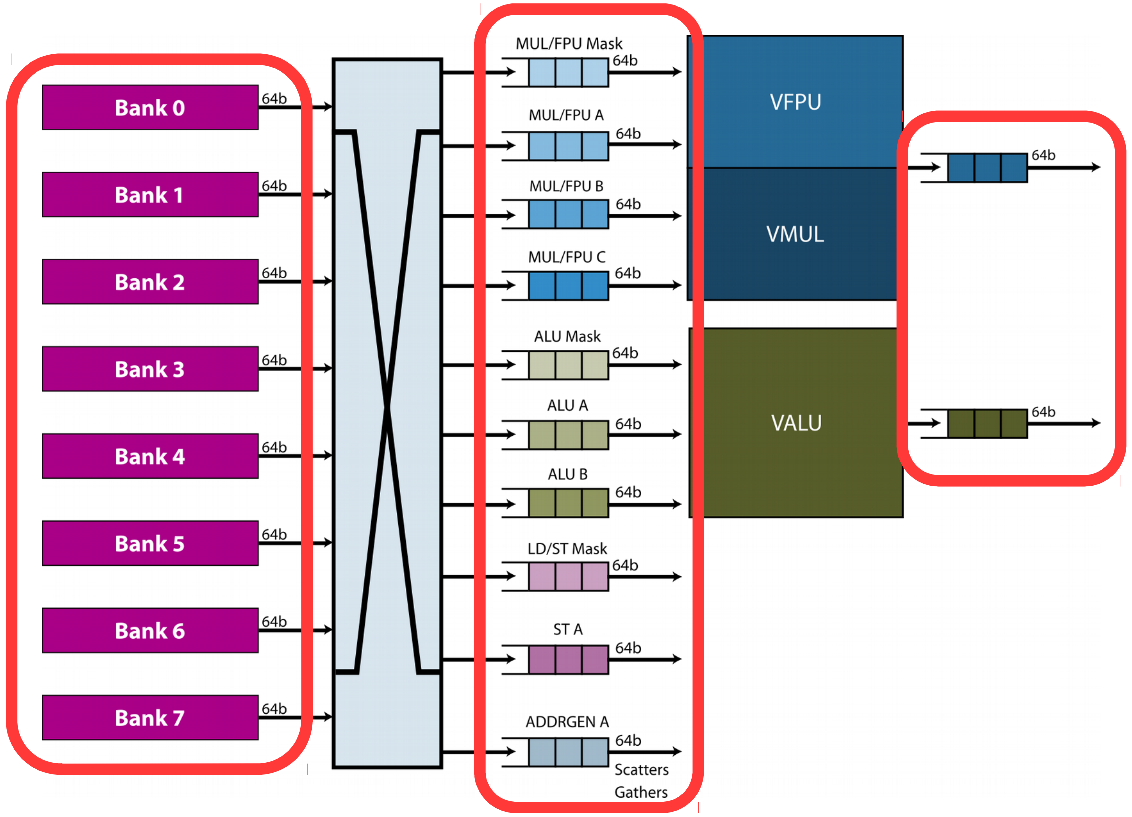
- **Lane sequencer** tracks in-lane operations
- **8 banks per lane**
- **Multiple requests** to *VRF* → Need for an **arbiter**
- Multiple **operand queues** to feed the FUs
- **Direct write-back** from the FUs





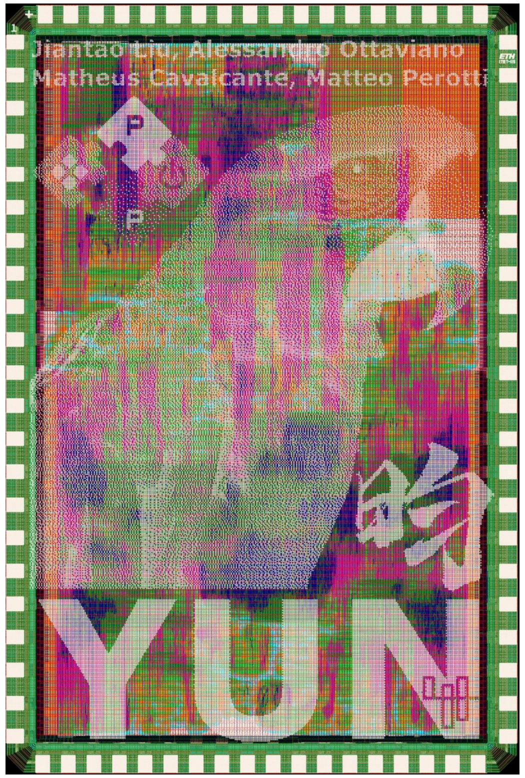
In lane - Vector Register File, Functional Units

- **8 banks per lane** → 8 single-ported SRAM
- **Crossbar** to feed the operand queues
- **2 result queues** (VMUL and VFPU share one)
- **Arbiter** for bank conflicts



What has been done

- Memory coherence between Ara and Ariane
- **Support** to **RVV 0.10,** implementing:
    - **Mask Unit** (different mask layout from RVV 0.5 to 0.10)
    - **Slide Unit** (crucial for `conv2d`)
    - Constant-strided vector loads and stores
    - Misaligned vector memory operations
    - Narrowing and widening arithmetic and floating-point instructions
- Push **decoding** into **Ara** for increased flexibility
    - Increase decoupling between the core and Ara
    - More RV extensions-agnostic Ariane
- Yun TO


Jiantao Liu, Alessandro Ottaviano
Matheus Cavalcante, Matteo Perotti
YUN

Currently ongoing

- Implementing:
  - Move whole-register
  - Whole-register memory operations
  - Mask memory operations
  - Integer reductions

Missing instructions

- Scatter-gather
- Fault-Only-First loads and stores
- Fixed-Point arithmetic
- Specific mask instructions (vpopc, vfirst, viota, vid, vmsbf, vmsif, vmsof)
- Gather/compress registers
- Floating-point reciprocal, reciprocal square root, round towards odd
- Move scalar elements from Ara to Ariane
- Floating-point reductions

SW environment – Current status

- **Compiler support**
  - LLVM + SPIKE support for RVV 0.10
  - Compiler support is not optimized
- **Refactoring our benchmark kernels**
  - `Imatmul, fmatmul, conv2d`
  - Manually vectorized
  - Intermix C + RVV assembly
    - Usage of intrinsics is possible
- **Verification**
  - Extending *riscv-tests* with ad-hoc *RVV* tests and macros

Verification

- **Support for 2, 4, 8, 16 lanes**
  - Verilator used to test all configurations
  - Check GitHub for more details