semidynamics

Semidynamic's RISC-V RVV1.0 Out-of-order Vector Unit

Roger Espasa, PhD, CEO & Founder

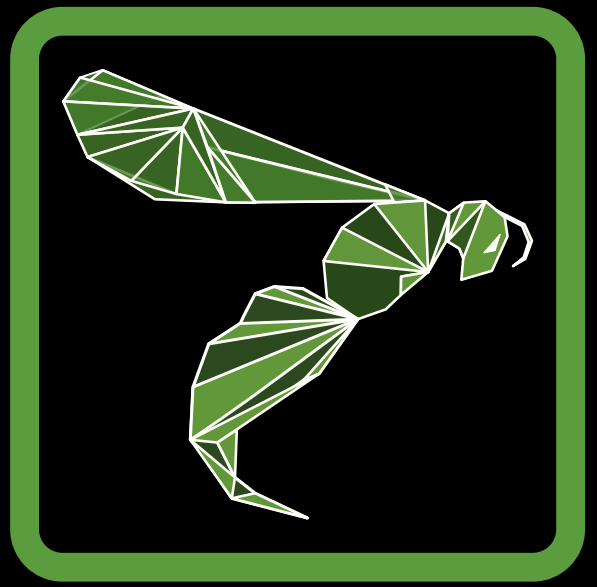
Semidynamics

semidynamics

Our RISC-V IP

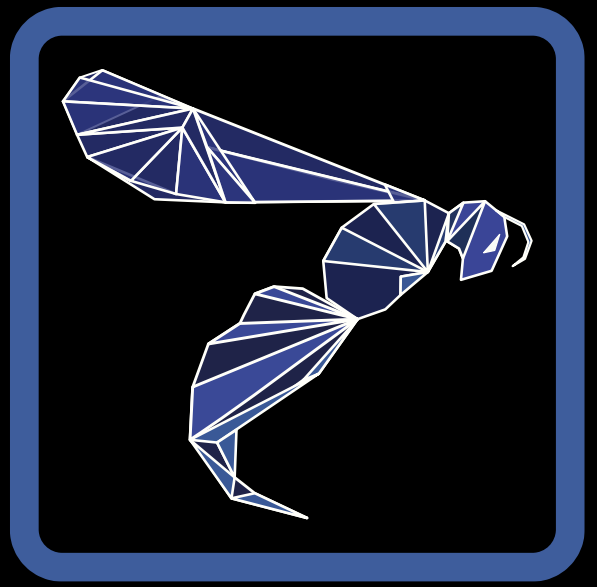
Avispado 222

2-wide **in-order** RISCV64GCV
AXI and CHI



Atrevido 222

2-wide out-of-order RISCV64GCV
AXI and CHI

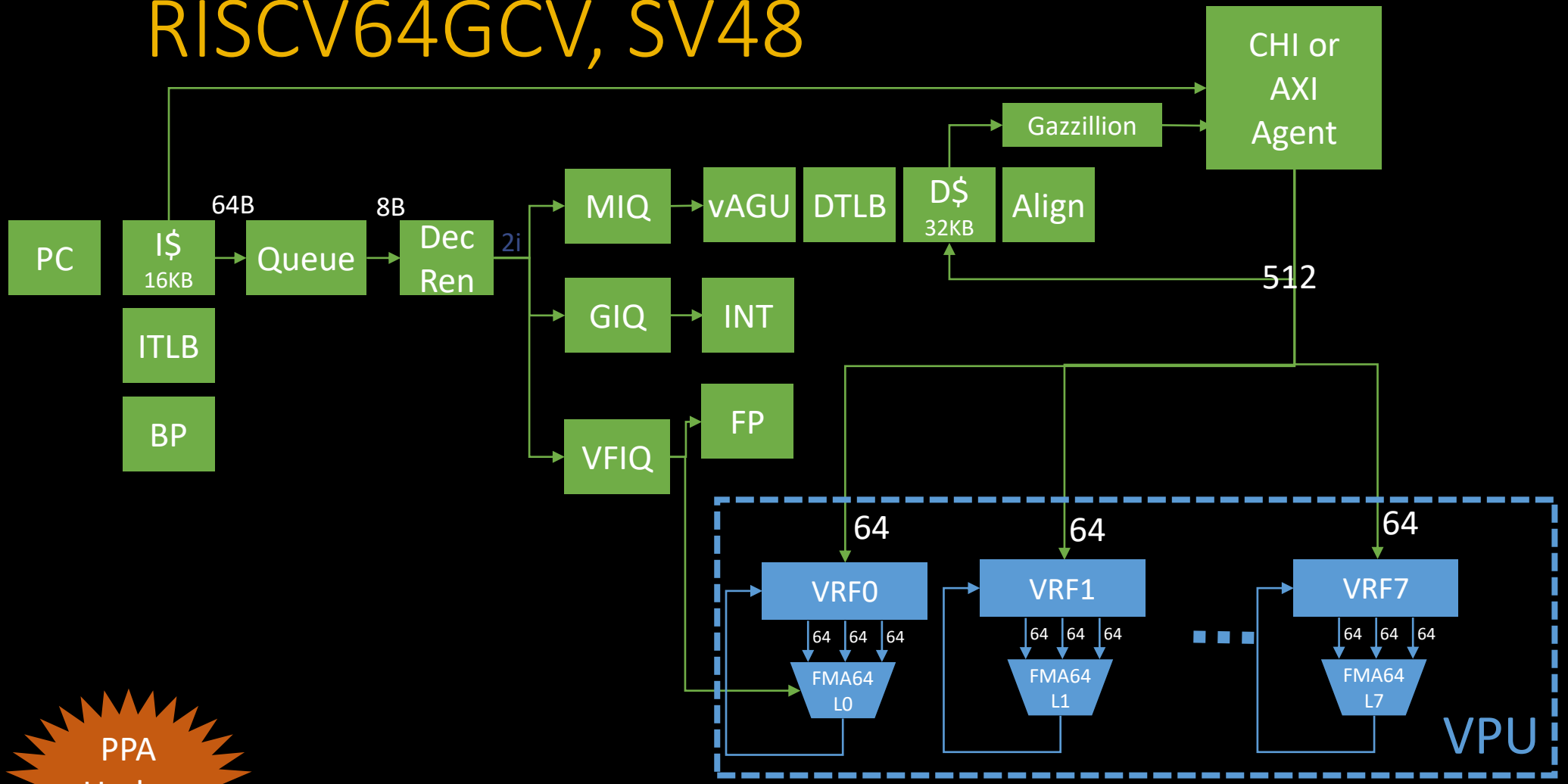


VPU 822

8-lane RVV1.0

semidynamics

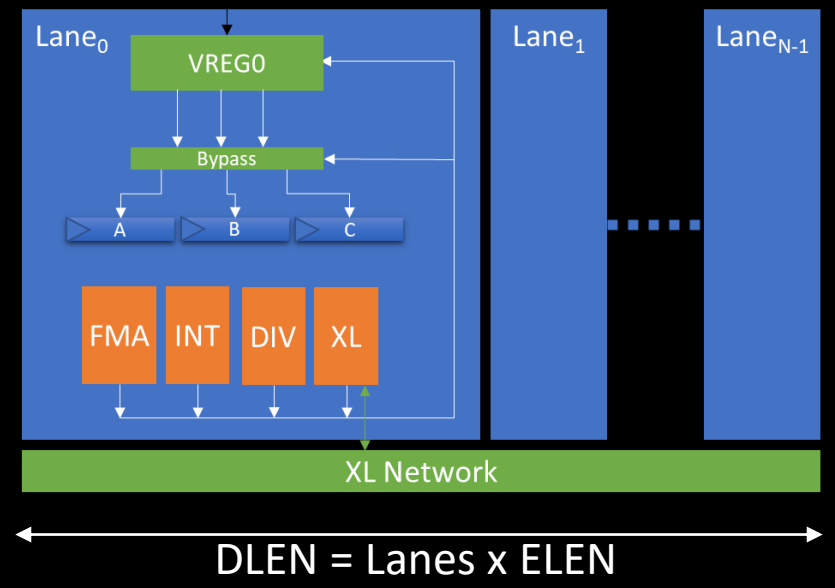
ATREVIDO 222 with VPU RISCV64GCV, SV48



- Linux SV48, SV39
- OOO
- CHI Coherency
- Unaligned Support
- RVV1.0

PPA Under NDA

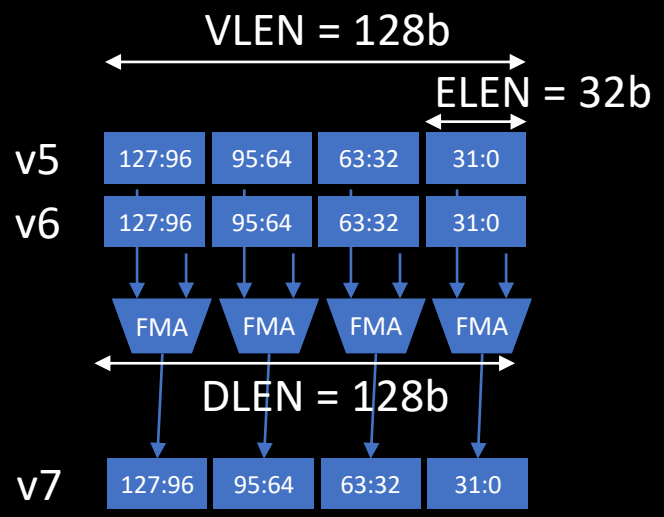
Lane Details



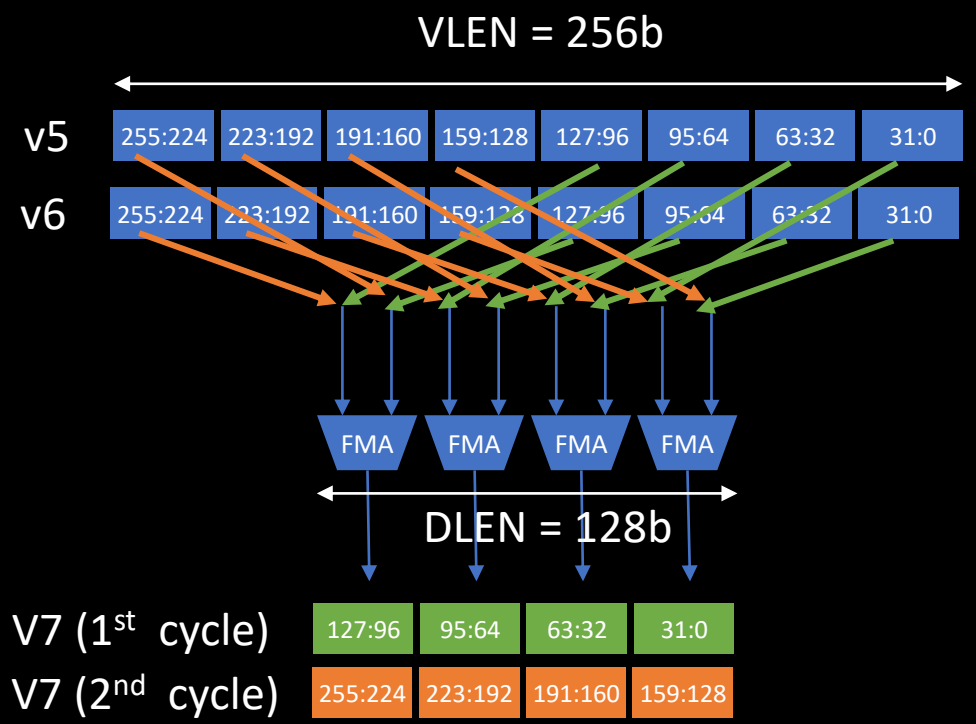
- Implements the RISC-V Vector 1.0 Specification
  - Including lmul, segmented loads
- Lane based organization
  - Full cache-line bus from D-cache
  - Units per lane: FMA, INT, DIV, XL: Cross-lane (rgather, …)
  - Full masking support
  - Fast cross-lane network for slide/rgather/compress/expand
- Ready for OOO
  - Supports vector register renaming
- Customizable settings
  - ELEN = maximum element size = 16, 32, 64
  - VLEN = vector register bits = from 128b to 4096b
  - DLEN = data path bits = from 128b to 512b

Customer's choice: ELEN, VLEN & DLEN

Data Types Supported	ELEN		
	16	32	64
FP64, INT64			☑
FP32, INT32		☑	☑
FP16, INT16	☑	☑	☑
BF16	☑	☑	☑
INT8	☑	☑	☑



VLEN == DLEN is SIMD



VLEN > DLEN is VECTOR

We support VLEN/DLEN ratios of 1, 2 and 4

Why VLEN > DLEN?

- Large VLEN is good in general…
  - Extracts more “Data-Level-Parallelism” from your application (if available)
  - One vector instruction “sees ahead” VLEN/ELEN iterations in your loop
- Large DLEN, however, is quite expensive
  - Big FMAs are replicated…. Take a lot of area
- VLEN > DLEN is a very attractive option with the following bonuses
  - Only grows vector register file area (but not FMAs)
  - Helps cover FMA latency
    - Dependent back-to-back vector FMAs don’t experience bubbles
  - Helps reduce core power
    - The Fetch, Decode, Issue stages clock gate for many cycles while VPU is working
  - Helps better use available “ROB size”
    - If your core has, say, 32 inflight instructions, larger VLEN will extract more performance

Feeding the VPU... (assuming VLEN=512b, cache line=512b)

DENSE DATA?

(i.e., matmul)

Burst of 28 vloads touch…

28..42 cache lines!

SPARSE DATA?

(i.e., embedding)

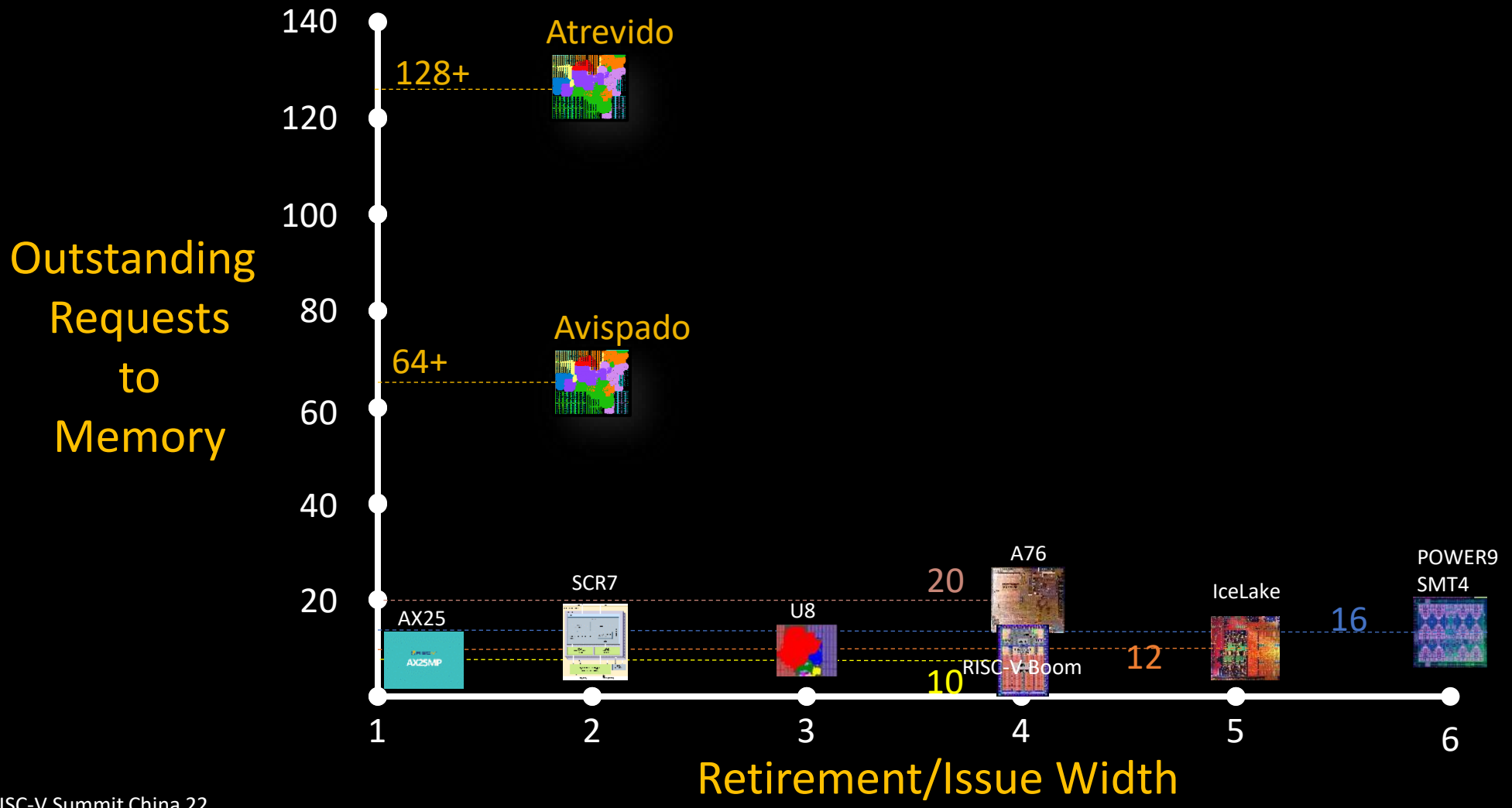
10 gathers @ SEW=32b touch

160 cache lines!

To keep the VPU fed, your core needs the ability to make lots of requests to memory

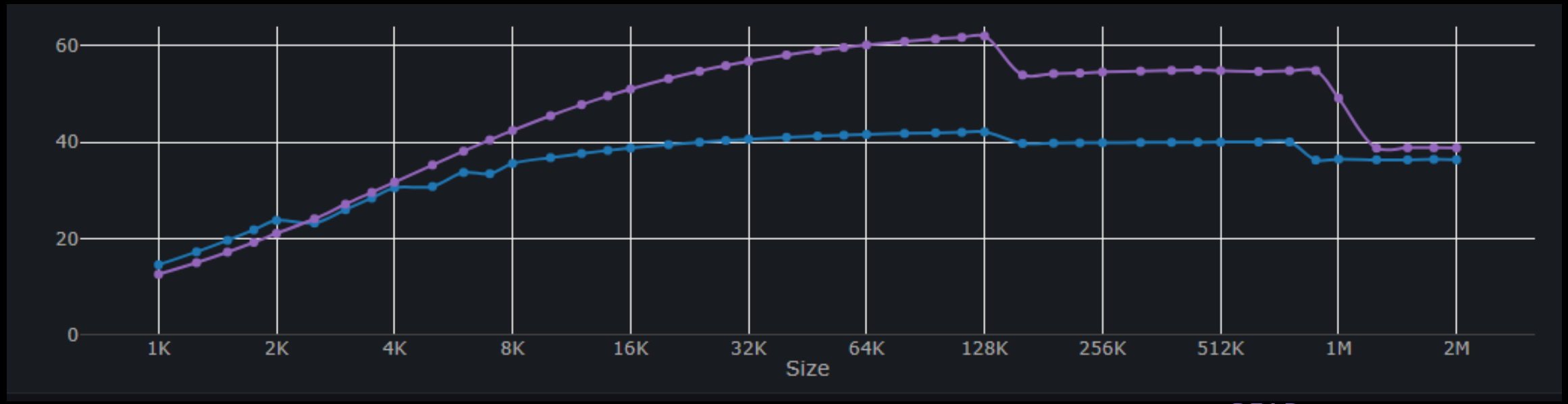
semidynamics

Our Secret Sauce to feed the VPU? Gazzillion Misses™



Gazzillion Misses™ incredibly good for RVV

Can you find a core out there capable of streaming data at over 60 Bytes/cycle?



Our VPU is Out-of-Order Ready

- Supports vector register renaming
  - Up to 64 physical vector registers (32+N for renaming)
  - Also renaming for “VL” and “VTYPE”
  - Supports “background copies” when needed by Renamer
  - Optimizes vector mask management to minimize cross-lane copying
  - Handles LMUL > 1
- Supports out-of-order data return from vector loads
- VRGATHER control unit handles N physical inputs under LMUL

semidynamics

European Processor Initiative

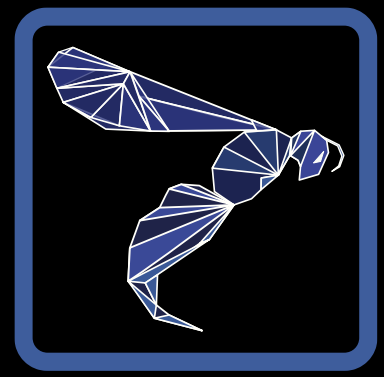
epi

THE EUPILOT

Thank you!

Avispado 222

Atrevido 222



VPU 822