Introducing SemiDynamics High Bandwidth RISC-V IP Cores

Roger Espasa, PhD

CEO

SemiDynamics

- European Based RISC-V IP Provider
  - HQ in Barcelona
  - Founded 2016
- Coming out of Stealth Mode today
- Proud participant in the European Processor Initiative (EPI)
  - RISC-V Acceleration core

Introducing 2 new RISC-V IP Cores

AVISPADO 220 

2-wide In-Order

Gazzillion Misses™

Open Vector Interface

ATREVIDO 220 

2-wide Out-of-Order

Gazzillion Misses™

Open Vector Interface

Available for licensing

What's a high bandwidth core?

A core with lots of outstanding memory requests

A core with Gazzillion Misses™

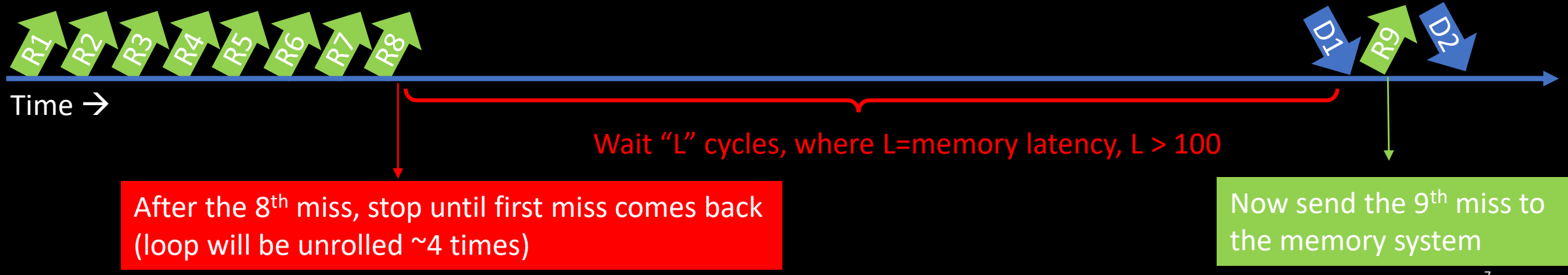
Example: Gathering Sparse Data with 8 outstanding requests

Original Program

```
while ( condition )
    load  a[index[i]]
    load  b[i]
    compute
    store c[i]
```

Stylized Assembly Code

```
top:
    load  index[i] -> x5
    load  (x5) -> x6
    load  b[i] -> x7
    compute x6, x7 -> x8
    store x8 -> c[i]
    loop to top
```



Example: Gathering Sparse Data with *Gazzillion Misses™*

Original Program

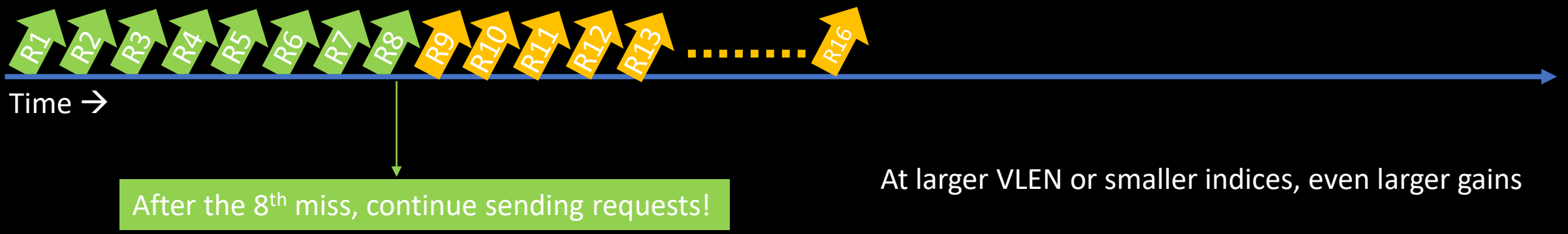
```
while ( condition )
    load  a[index[i]]
    load  b[i]
    compute
    store c[i]
```

Stylized Assembly Code

```
top:
    load  index[i] -> x5
    load  (x5) -> x6
    load  b[i] -> x7
    compute x6, x7 -> x8
    store x8 -> c[i]
    loop to top
```

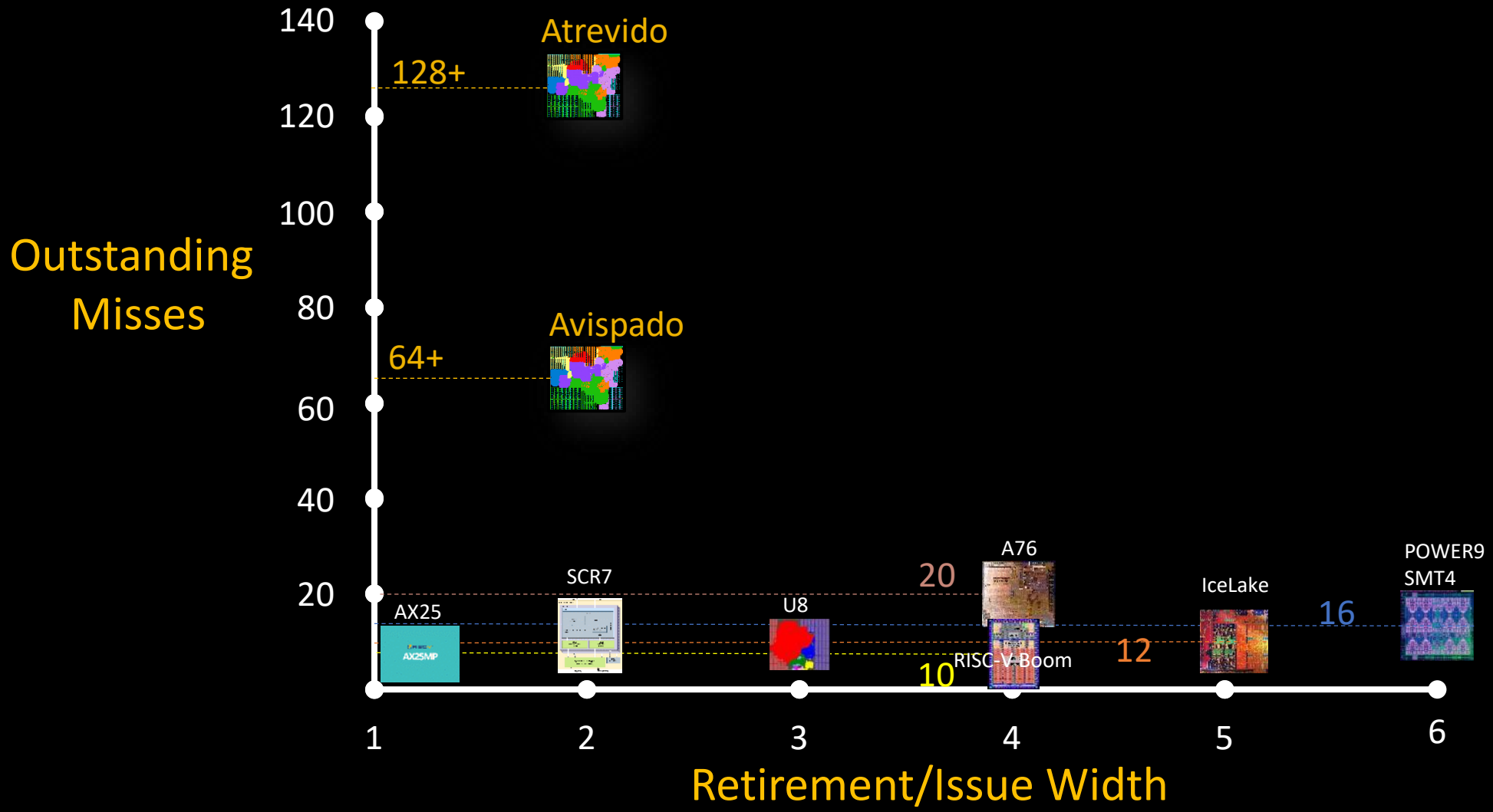
Another Example: Vector Gather with *Gazzillion Misses™*

- Consider the RISC-V "vlxei32.v" gather instruction
- Assume VLEN=512b
- Instruction will request 16 addresses (assume to different cache lines)



At larger VLEN or smaller indices, even larger gains

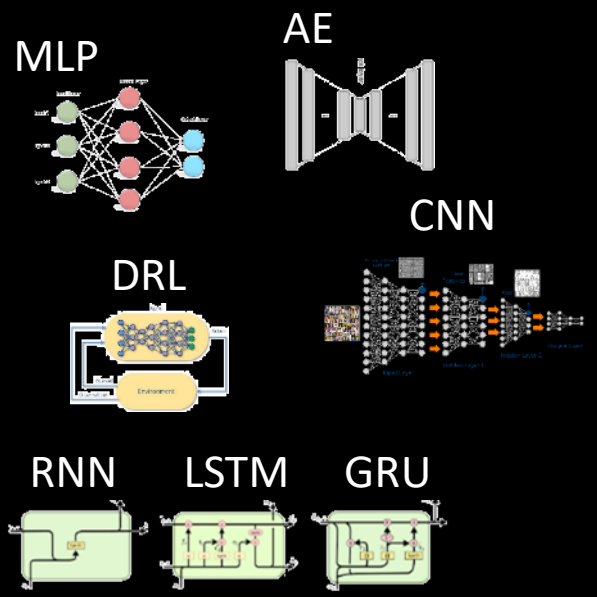
Comparison to other cores



Gazzillion Misses™ good for…

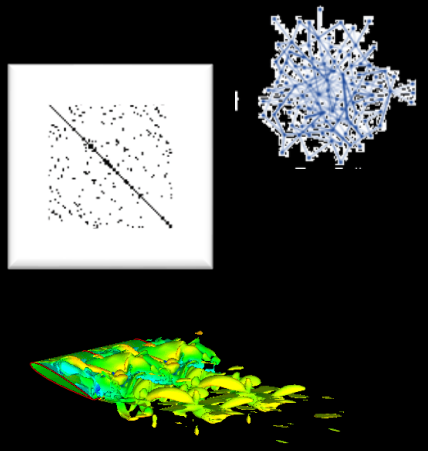
Machine Learning

Recommendation Systems



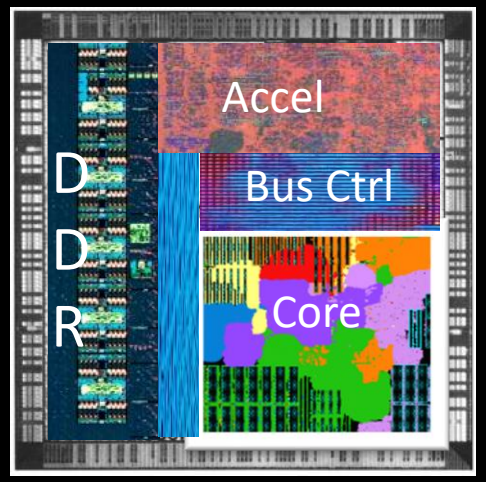
Key-Value Stores

Sparse Data / HPC

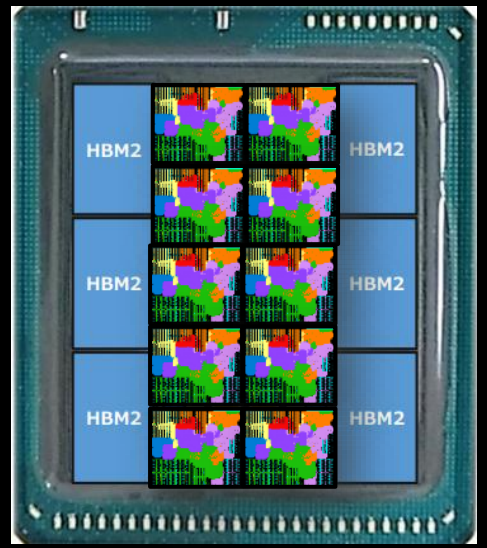


Gazzillion Misses™ good for SoCs...

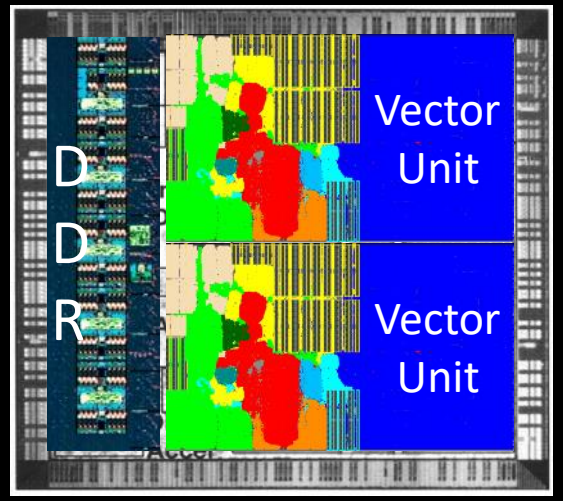
With Limited SRAM/Cache



High Bandwidth/ Streaming



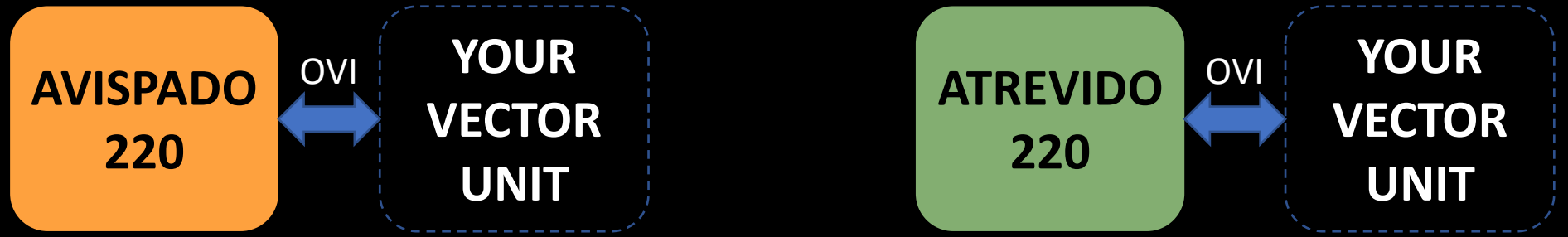
With Vector Units



What's the Open Vector Interface?

Open Vector Interface (OVI)

- Open Source Interface
  - Download definition from GitHub
- Connect your own Vector Unit to SemiDynamics' Cores
  - Vector Loads, Stores, Gather, Scatter taken care of for you
  - Virtual Memory, PMA, Coherency taken care for of you
  - Supports both in-order and out-of-order execution
- You can focus on the computation capabilities of your target market



Signals between Core and VPU

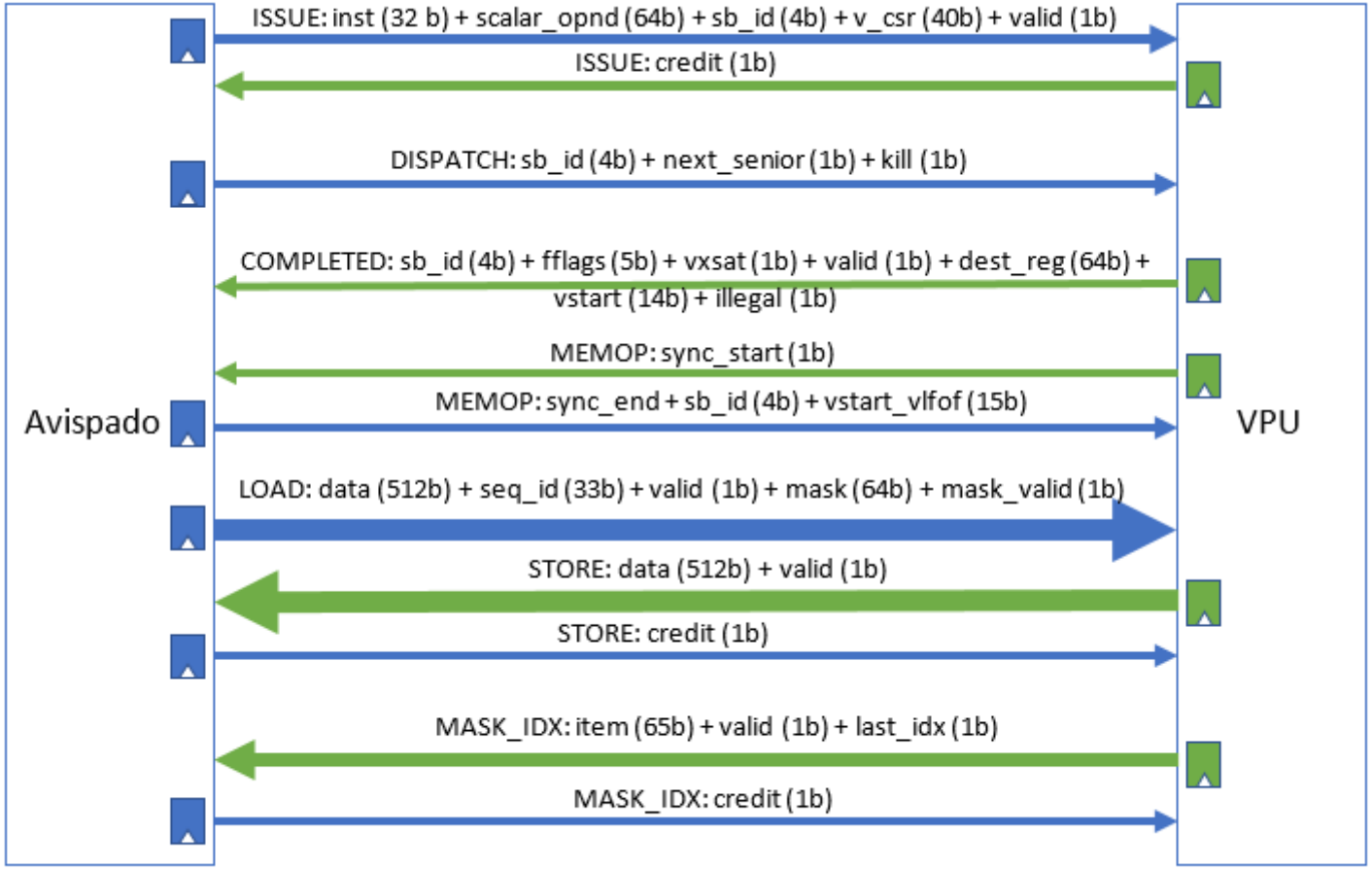


Figure 1. Signals and buses connecting Avispado to the VPU

- ISSUE signals
  - From core to VPU
  - Credited
- Completion signals
  - From VPU to core
- Load Data
  - From core to VPU
- Store Data
  - From VPU to core
  - Credited
- Mask information
  - From VPU to core
  - Credited

A couple examples from the Open Spec:

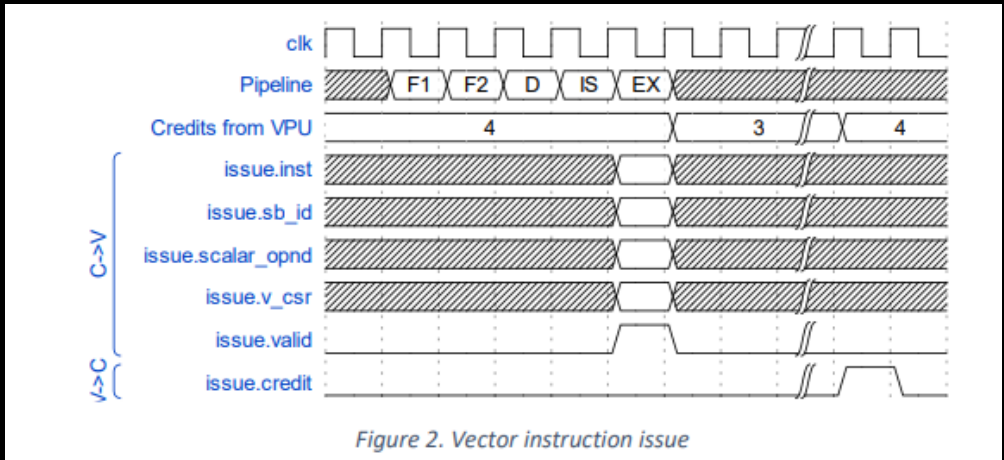


Figure 2. Vector instruction issue

- Core has 4 credits from VPU
- Sends next vector instruction to VPU, decrements credits

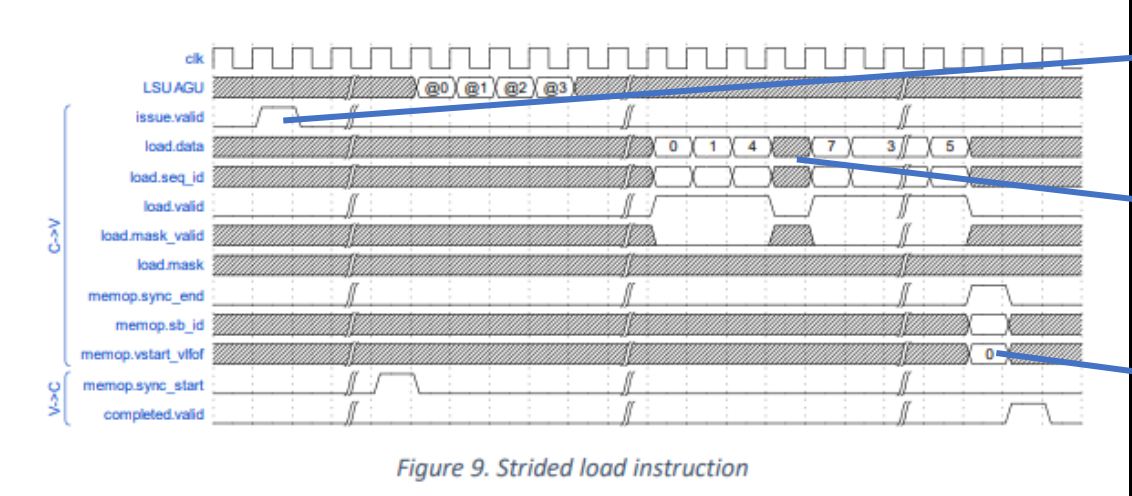


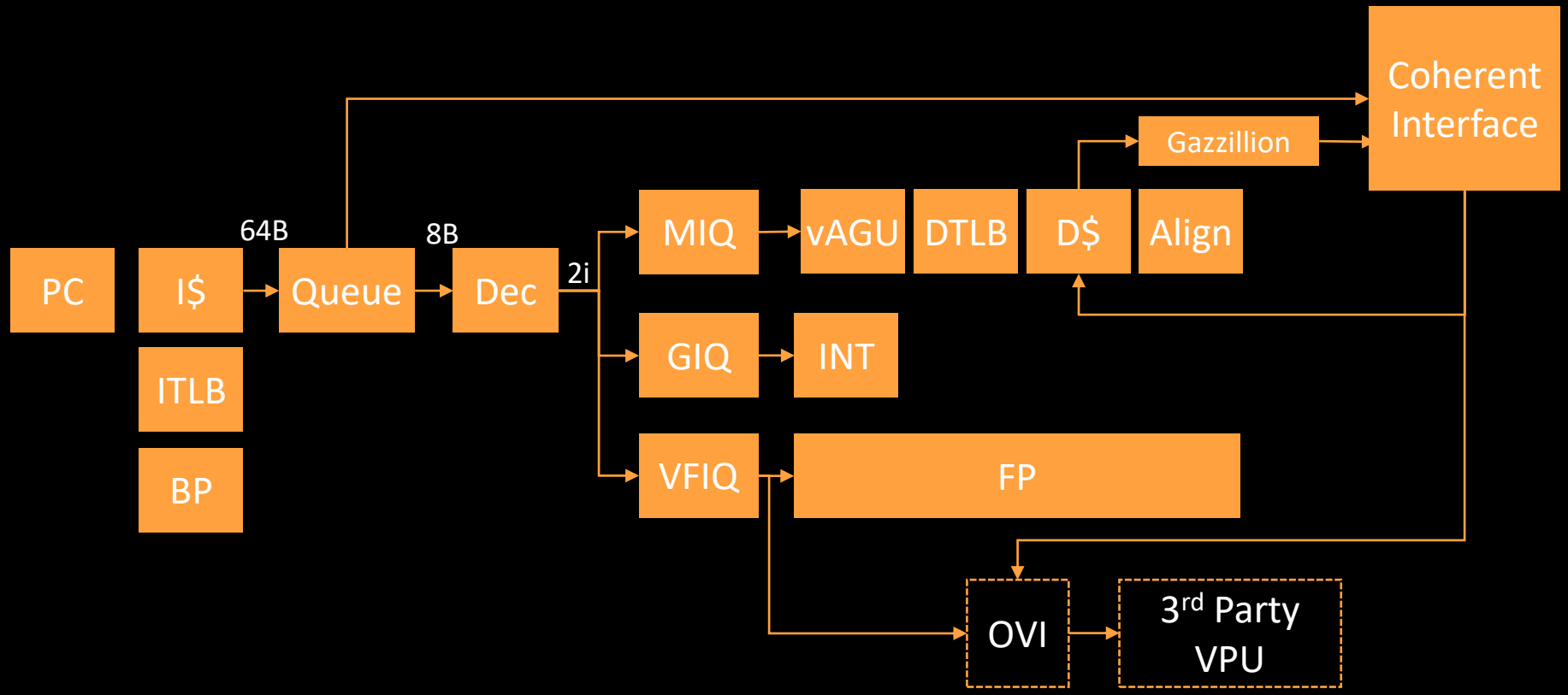
Figure 9. Strided load instruction

- VPU / Core handshake to start vload
- Load data returning out of order
- Completion signal

Core Details

AVISPADO 220

RISCV64GCV

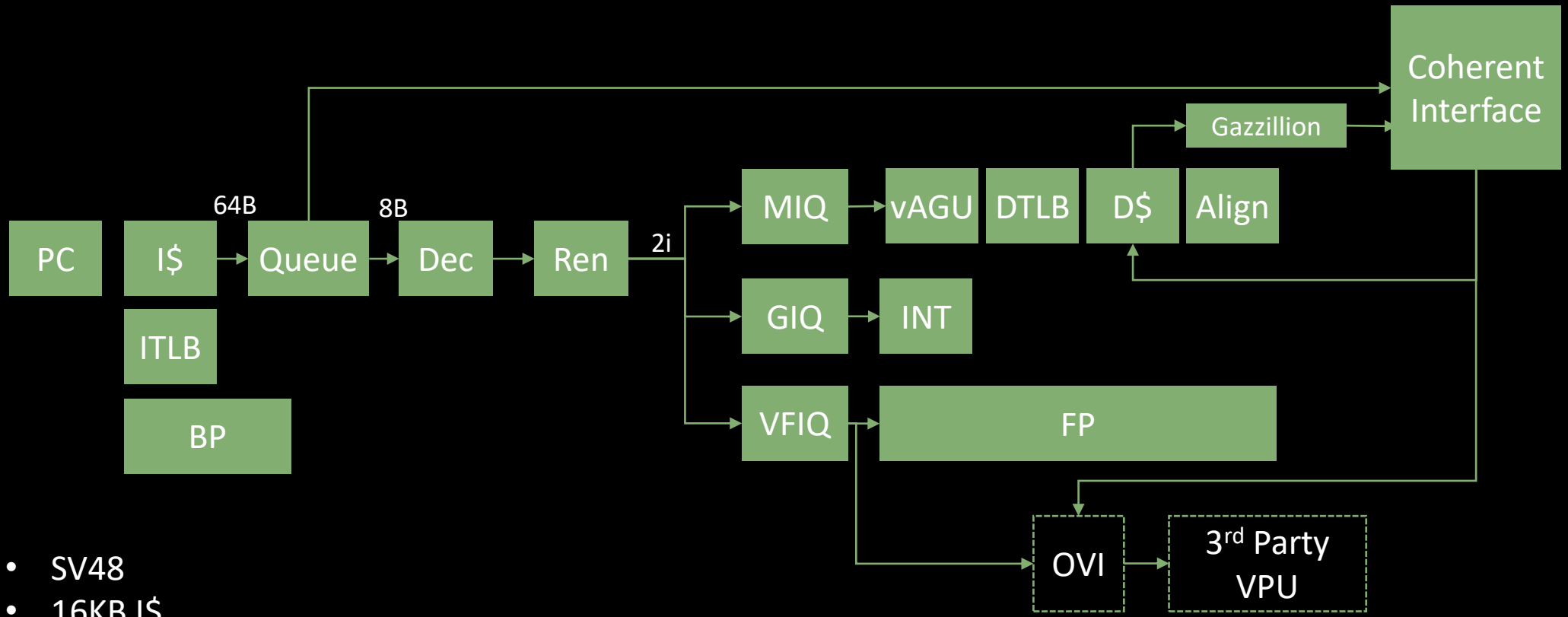


- SV48
- 16KB I$
- Decodes upcoming V1.0 vector spec
- 32KB D$
- Full hardware support for unaligned accesses
- Coherent (TL, ACE, CHI)

Available for licensing

ATREVIDO 220

RISCV64GCV



- SV48
- 16KB I$
- Decodes upcoming V1.0 vector spec
- 32KB D$
- Full hardware support for unaligned accesses
- Coherent (TL, ACE, CHI)

- Larger BP
- Renaming
- Retirement Logic

Available for licensing

Customizable upon request

- Cache Size / Associativity
- Coherent Interface
- In-flight instructions
- OVI buffers
  - Vector Store Data Buffer
  - Masking credits

Summary

AVISPADO 220 

2-wide In-Order

Gazzillion Misses™

Vector Spec 1.0

Open Vector Interface

ATREVIDO 220 

2-wide Out-of-Order

Gazzillion Misses™

Vector Spec 1.0

Open Vector Interface

IP cores available for licensing

Thank you!

www.semidynamics.com

info@semidynamics.com