Boosting AI on Semidynamics RISC-V Cores with Custom Tensor Instructions

RISC-V Summit North America
2023.11.07

Roger Espasa,
Founder & CEO

semidynamics

About Semidynamics

Semidynamics, founded in 2016, is a **100% European** supplier of RISC-V IP cores, HQ in **Barcelona**, specializing in **customization** of **high bandwidth high performance cores with vector units** for **tailored projects**

Experts in open core surgery

semidynamics

Our RISC-V CPU Core IP Families

Atrevido

无所畏惧　Fearless　용감한

2, 3 or 4-wide **out-of-order**
RISCV64GCV AXI and CHI

Avispado

机智　Smart　똑똑한

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

World's first, **fully customizable**, 64-bit RISC-V cores for ultra fast, big memory applications, optimized for a companion RISC-V **vector unit**

Unique tailor-made PPA solutions include customer's secret sauce for product differentiation and IP protection.

semidynamics

AI/ML Everywhere

- State-of-the-art AI/ML models are huge
  - Billions of parameters
  - Require trillions of operations per second
    - Mainly in convolutions and fully-connected layers (matrix multiplication)
- RISC-V for AI/ML
  - Vector extensions represent a large performance boost over scalar…
  - but working with matrices provides significant benefits
- Our solution
  - Extend RISC-V Vector Unit with tensor instructions
  - Include Tensor Unit on top of the Vector Unit

Tensor Unit

- Tensor unit **directly-connected** to the Vector Unit
  - No DMA, uses regular vector load and store instructions
- Very **easy programming** model, using 3 new instructions
  - vmxm : matrix-multiply instruction
  - vlrs : vector load with row stride
  - vsrs : vector store with row stride
- **Linux-ready**
  - No kernel modifications needed; state saved through vector registers
- Supports **INT8, INT16, FP16, BF16**
  - Accumulations at 32 bits (FP32/INT32) to avoid precision loss

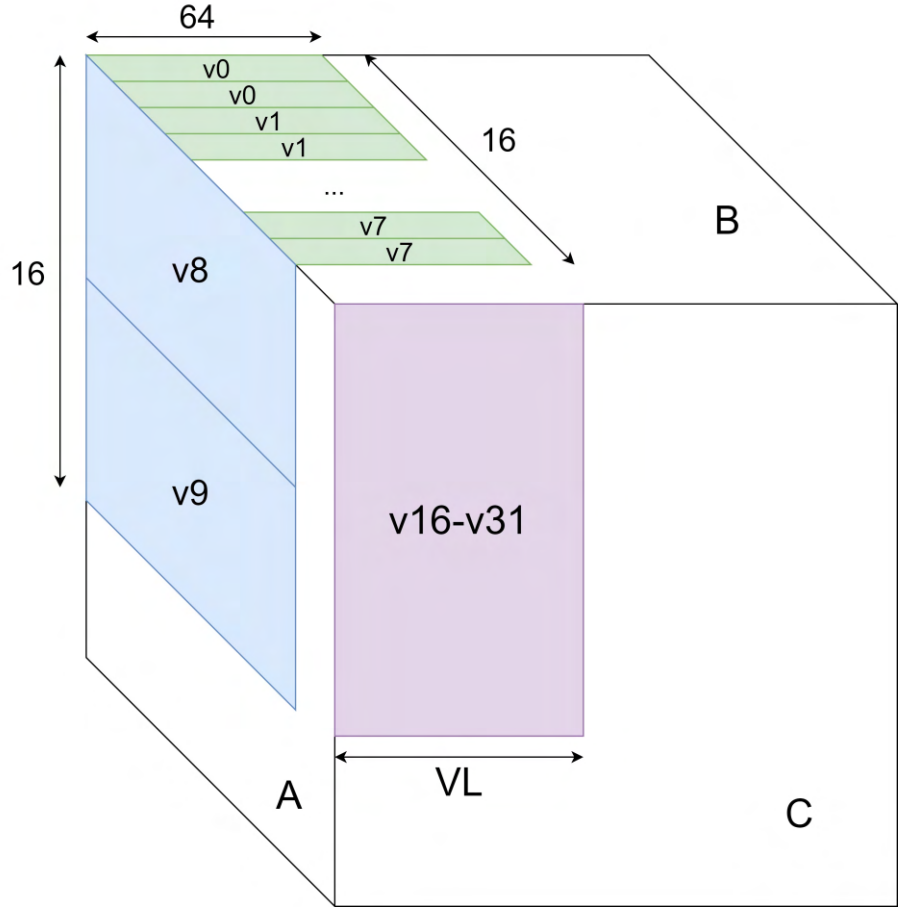
New Instructions:

- **vlrse** vd, (rs2), rs1
  - rs2: base address
  - rs1: number of elements per row and row stride in bytes
- **vsrse** vd, (rs2), rs1
  - rs2: base address
  - rs1: number of elements per row and row stride in bytes

- **vmxm{acc}**
  - C = A x B + C
  - A matrix in v8-v9
  - B matrix in v0-v7
  - C matrix in v16-v31
  - Element width from vtype

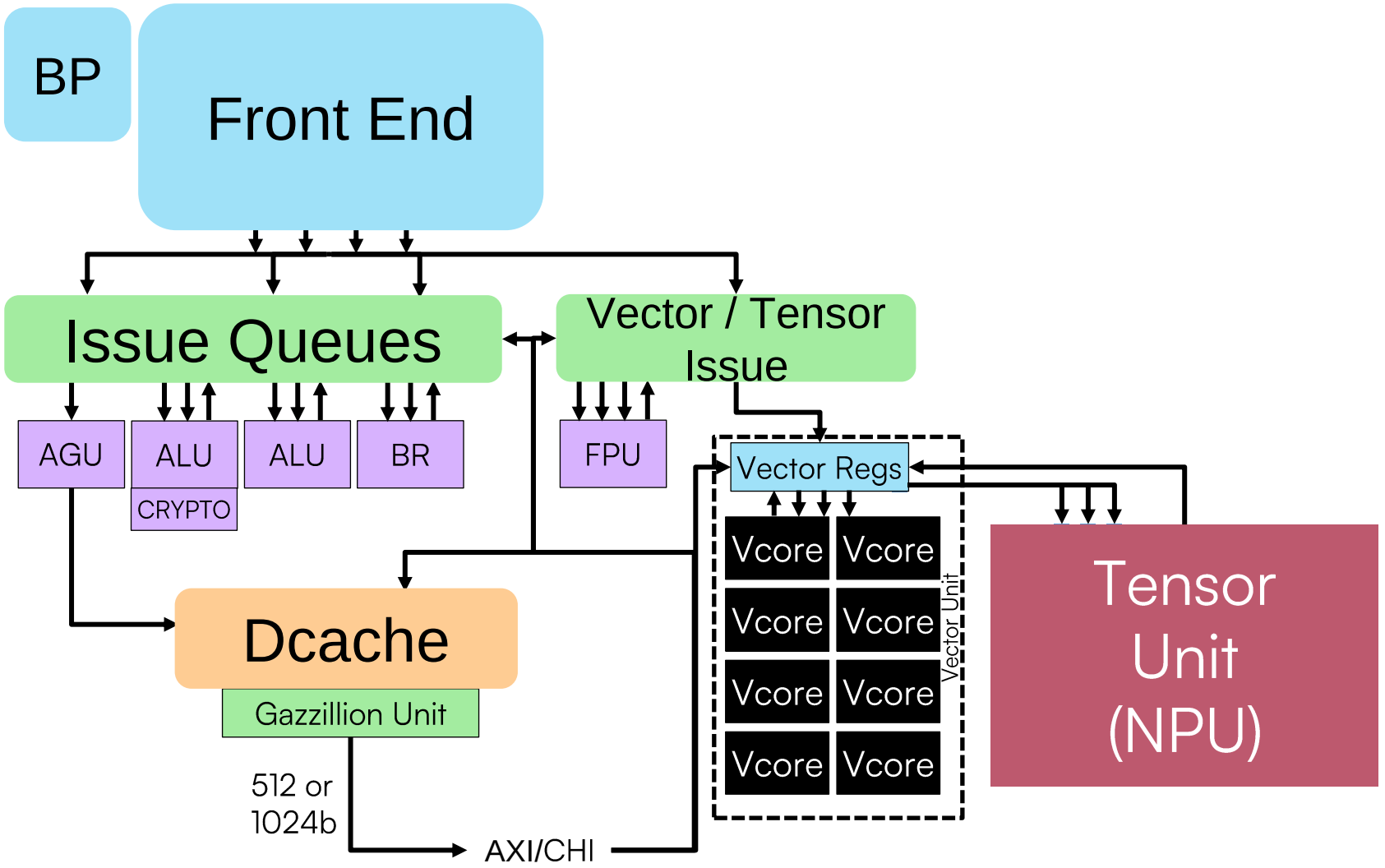
We really mean easy : 8-instruction loop

```
# C tile pre-loaded into v16-v31
loop: vsetvli  zero, t4, e16, m2, ta, ma
      vlrs16   v8, (a0), t1
      addi     a0, a0, 32
      vsetvli  zero, t5, e16, m8, ta, ma
      vlrs16   v0, (a1), t2
      add      a1, a1, t3
      vfmxmacc v16, v8, v0
      bltu     a1, t6, loop
# Store C tile (v16-v31) back to memory
```



semidynamics

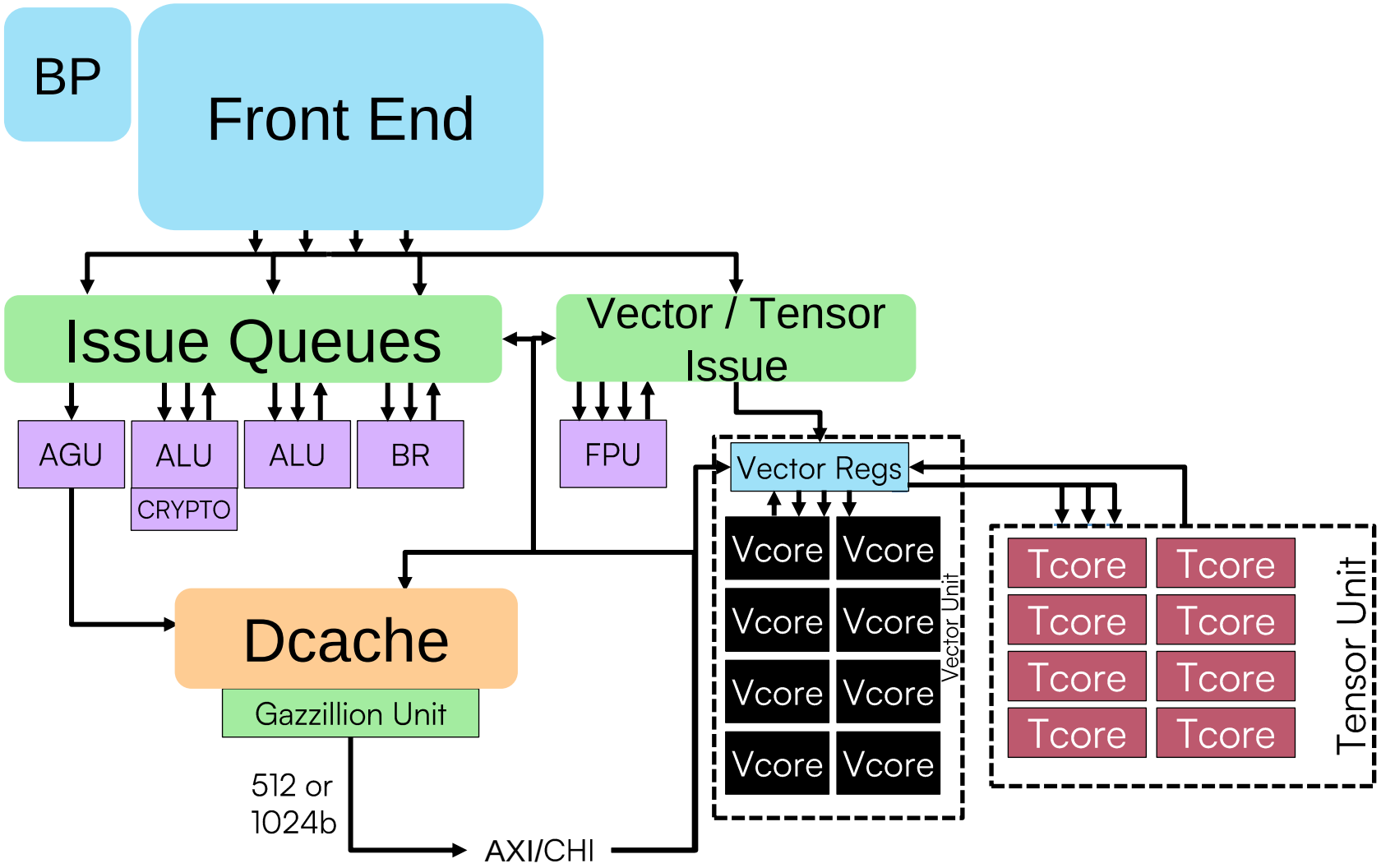
Atrevido 423-V8 + Tensor Unit



- Input data and weights stored in memory
- Fetched on demand into the Vector Register File
- Gazzillion Misses™ allows efficient data fetching from memory

semidynamics

Atrevido 423-V8 + Tensor Unit

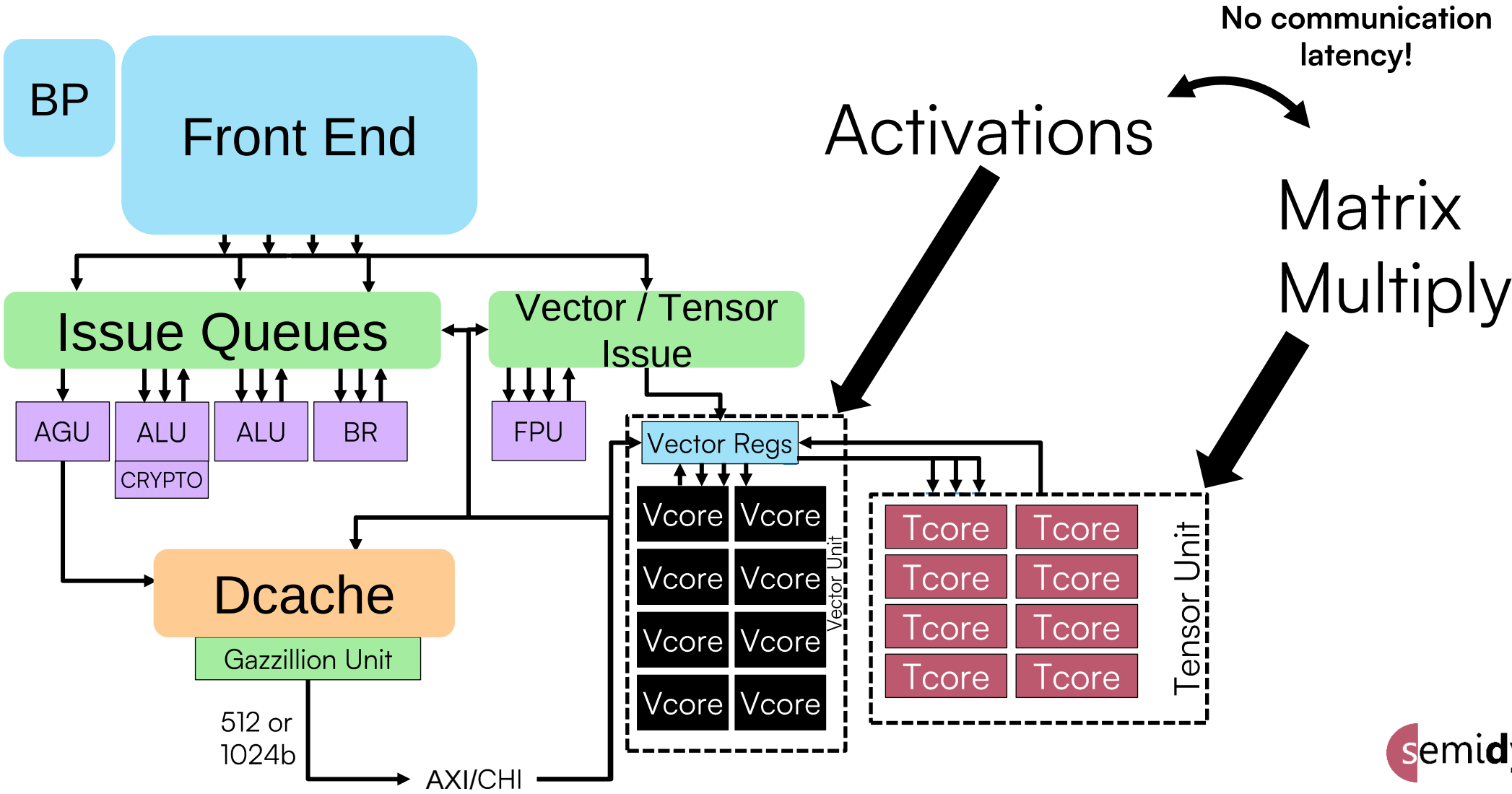


Each Tensor Core:

- Supports **INT8, INT16, FP16, BF16**
- Contains 32 INT8 MAC units
- Mixed precision
- Accumulations at 32 bits (FP32/INT32) to avoid precision loss

semidynamics

Atrevido 423-V8 + Tensor Unit

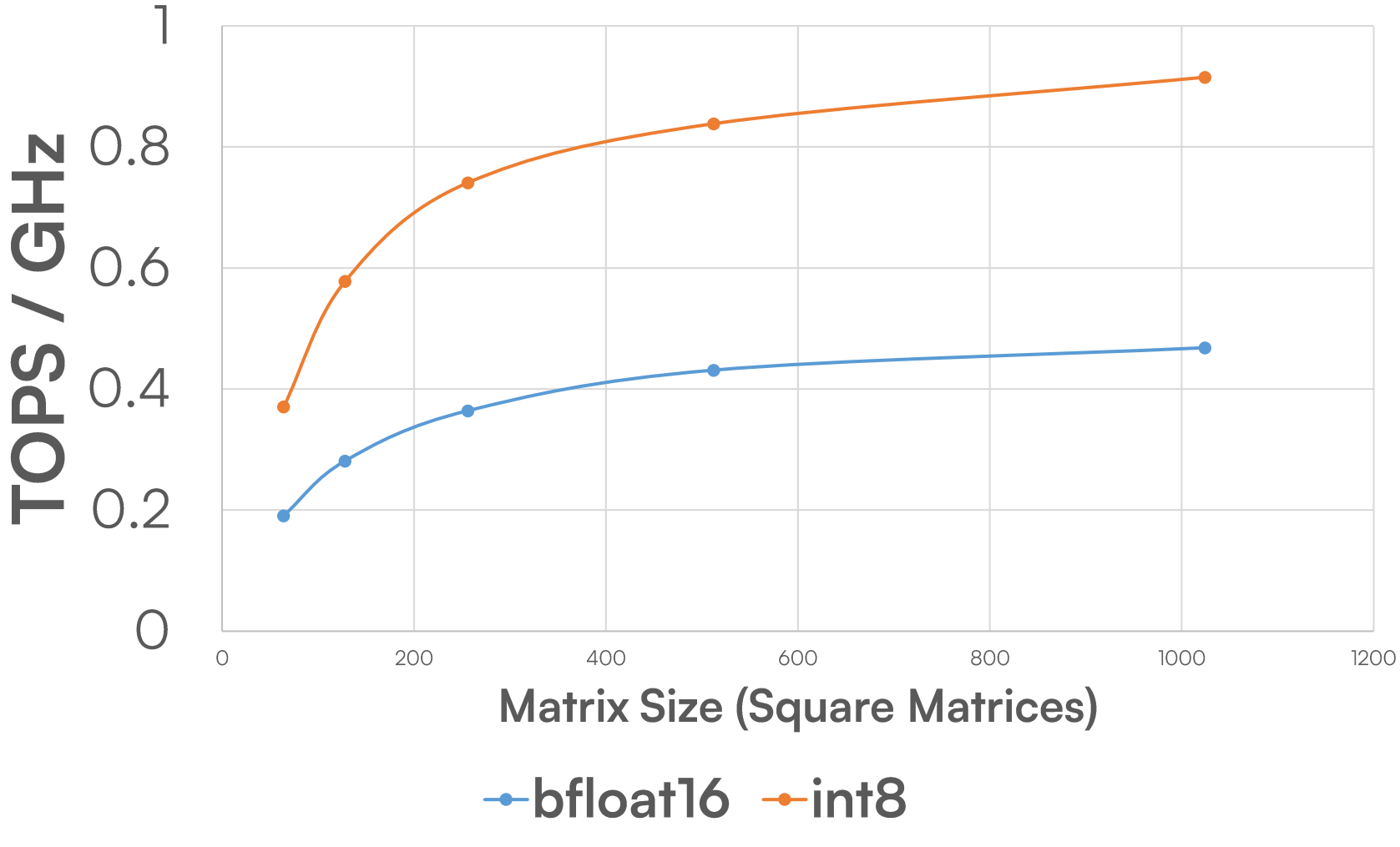


semidynamics

Tensor Unit Performance
(scales with VLEN and Frequency)

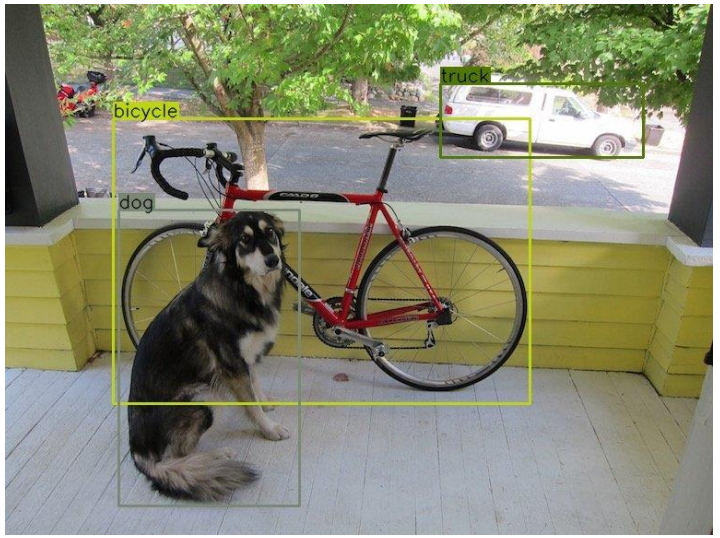
Product	T4	T8	T16	T32
VLEN	512	1024	2048	4096
MLEN (bus width)	512	512	512	1024
Tensor Cores	4	8	16	32
INT8 TOPS/GHz	0.25	0.5	1	2
INT16 TOPS/GHz	0.125	0.25	0.5	1
BF16 TOPS/GHz	0.125	0.25	0.5	1
FP16 TOPS/GHz	0.125	0.25	0.5	1

Matrix Multiplication on T16 (1 Tops peak)

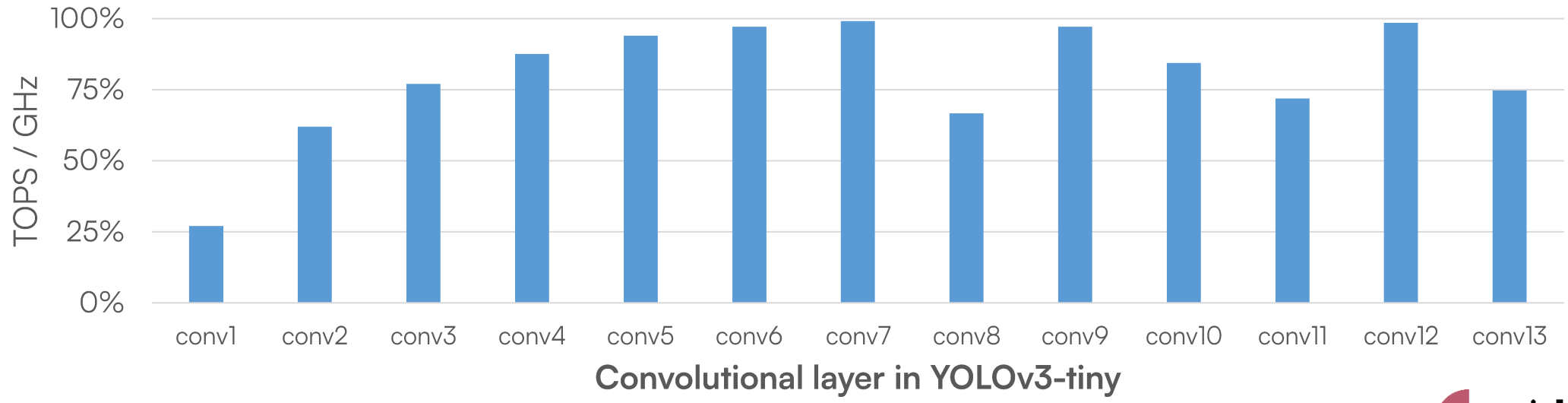


semidynamics

YOLOv3-tiny: 33 FPS



- Performance at 1GHz
  - ATV4+Vector Unit (bf16): 5.61 FPS
  - ATV4+Vector Unit+Tensor Unit (bf16): 33.03 FPS
  - Real-time performance with one Tensor Unit



semidynamics

Conclusions

- Custom **tensor instructions** on top of RISC-V V 1.0
  - No additional architecturally-visible state
  - Reuse existing vector register file
- Easy programming
  - AI/ML codes can easily exploit the new tensor instructions
- High performance
  - **Over 0.9 INT8** for matrix multiplication at 1GHz
  - **33FPS** for YOLOv3-tiny at 1Ghz