CICC

IEEE Custom Integrated Circuits Conference

A Low-Power 1.2 TFLOPS/W Merge-Split RVV Processor for Mixed Scalar-Vector Workloads

***Mattia Sinigaglia**, Marco Bertuletti, Gianmarco Ottavi, Amirhossein Kiamarzi, Matteo Perotti, Giuseppe Tagliavini, Luca Benini, and Davide Rossi*

Ph.D graduate

Energy Efficient Embedded System (EEES), MICREL Lab,

Viale Carlo Pepoli 3/2, 40123, University of Bologna, Italy

PULP Parallel Ultra Low Power

21 April 2026, Seattle

SSCS IEEE SOLID-STATE CIRCUITS SOCIETY™ | IEEE

Outline

- **Heterogeneous Embedded Computing**
  - Mixed Scalar-Vector workloads
  - Multi-Core Vector architecture's Limitations
- **Reconfigurability**
  - The proposed solution
    - Split-Mode
    - Merge-Mode
  - Hardware Architecture
    - Reconfiguration & Merge Interface
    - Vector Register File Layout
- **The Buckbeak SoC**
  - Physical Implementation
  - Performance and Energy Efficiency results
  - Impact of SoC reconfigurability
- **SoA Comparison**

Heterogeneous Embedded Computing

Mixed Scalar-Vector workloads

Resource constrained autonomous embedded systems combine compute-intensive parallellizable workloads with supervision and control tasks that must be executed concurrently.

- **Data parallel workload**
  - Filters (FIR – IIR)
  - Transforms (FFT – iFFT- STFT)
  - Linear Alg
  - Perceptio
  - Estimatio

Flexibility + Programmability!

Drones

Rovers

Robots

- **Supervisory and control workload**
  - Scheduling/I-O (IRQ – DMA – QoS)
  - Monitors (Watchdog – Timeout – SanityChecks – FaultFlags)
  - Finite-State-Machines (ModeManager – Arming/Failsafe – Recovery – MissionLogic)
  - Control-Loops (Supervisor logic – Trajectory step – Inner-loop update – Motor control)

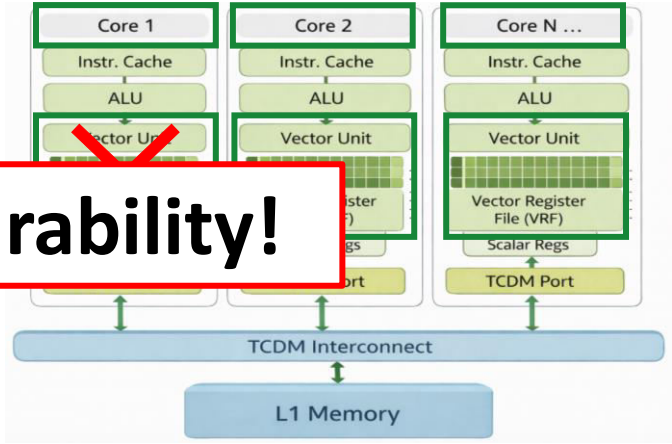
Heterogeneous Embedded Computing

Multi-Core Vector architecture's Limitations

On Mixed Scalar-Vector workloads conventional multi-core vector architectures suffer in redistributing the hardware resources[1][2].

- **Data parallel workload**
  - ✓ Excel in handling structured workloads exploiting data-level parallelism
  - × Suffer from
    facing low
- **Supervisory and control workload**
  - × Serialize the execution
  - × Allocate one vector-capable core to scalar tasks leaving vector resources idle.

Hardware Reconfigurability!



Example of a Multi-Core Vector Architecture

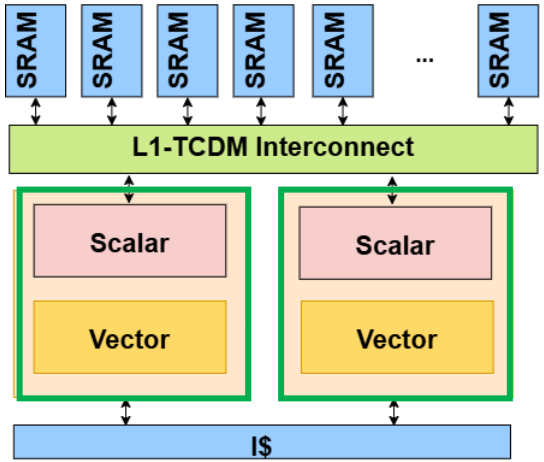
[1] Z. Zhang et al., "Occamy: Elastically sharing a SIMD co-processor across multiple CPU cores," in Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, ser. ASPLOS 2023.
[2] S. F. Beldianu et al., "Multicore-based vector coprocessor sharing for performance and energy gains," ACM Trans. Embed. Comput. Syst., vol. 13, no. 2, Sep 2013.

Reconfigurability

The Proposed Solution

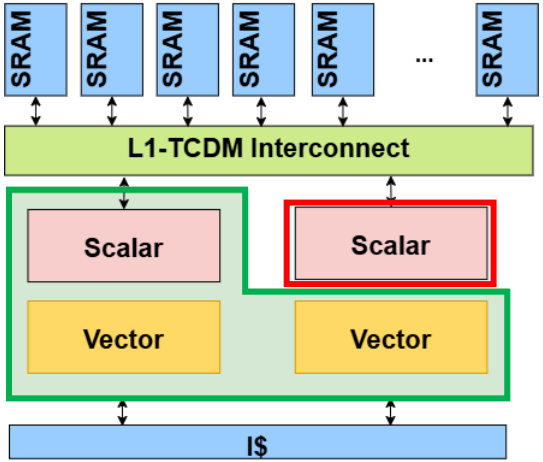
Dynamic remapping of multiple vector units to work in two operation modes.

Split-Mode (SM)



Each scalar core controls a Vector accelerator: The architecture works as a multi-core vector cluster.

Merge-Mode (MM)



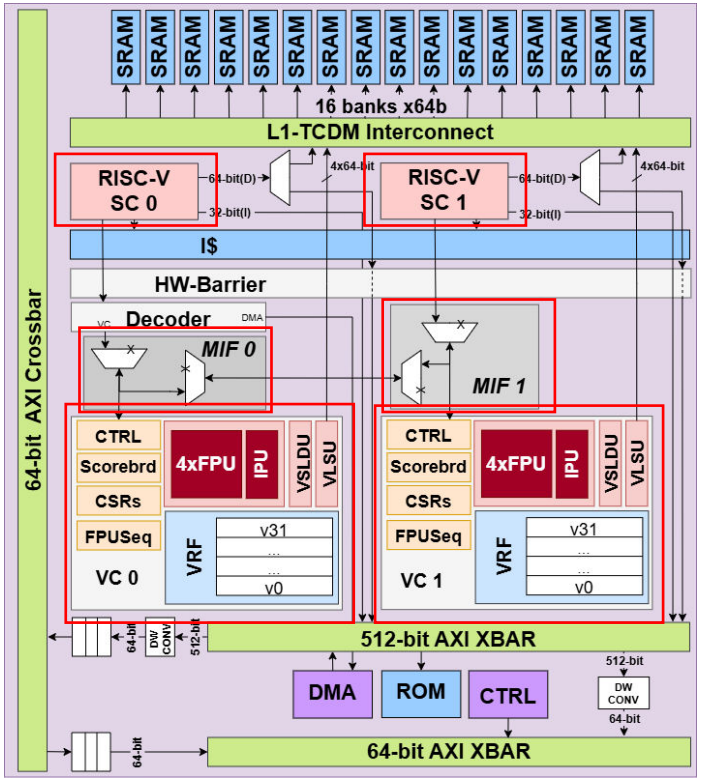
One Scalar core controls both Vector accelerators: The decoupled Scalar unit executes control tasks independently.

Reconfigurability

Hardware Architecture

This work enhances the open-source[1] Spatz Cluster[3] design with reconfigurability feature.

- 2×RISC-V 32-bit RV32IMAFD Scalar Cores (SC)
- 2×RVV1.0 Zve64d-based Vector Cores (VC)
  - 512b Vector Register File (VRF)
  - 4×64b Floating Point Units
  - Vector Load/Store Unit (VLSU)
  - Vector Slide Unit (VSLDU)
  - Integer Processing Unit (IPU)
- 128B L0 I$
- 512b DMA
- 128KiB L1 Memory
- 2× Merge Interfaces (MIFs)



[1] https://github.com/pulp-platform/spatz

[3] M. Perotti et al., "Spatz: Clustering Compact RISC-V-Based Vector Units to Maximize Computing Efficiency," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 7, pp. 2488-2502, July 2025.

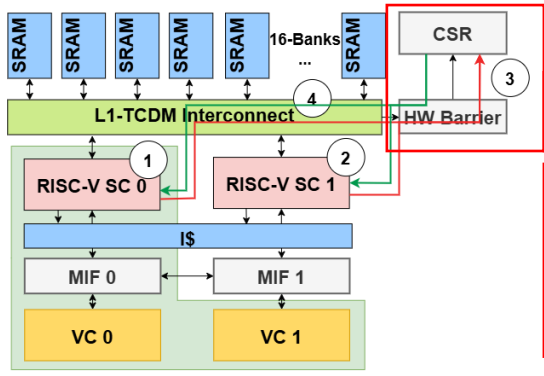
The Reconfigurability Feature

Reconfiguration & Merge Interface

Reconfiguration between Split Mode (SM) and Merge Mode (MM) is triggered at runtime by writing the *mode* Control and State Register (CSR) enabling fast adaptation to workload behavior.

- Each MIF defaults to SM configuration.
- In MM cross-core communication is established:
  - SC 0 controls both VCs → MIF 0 forwards vector instructions to MIF 1
  - SC 1 detach from its vector datapath

5 cycles!

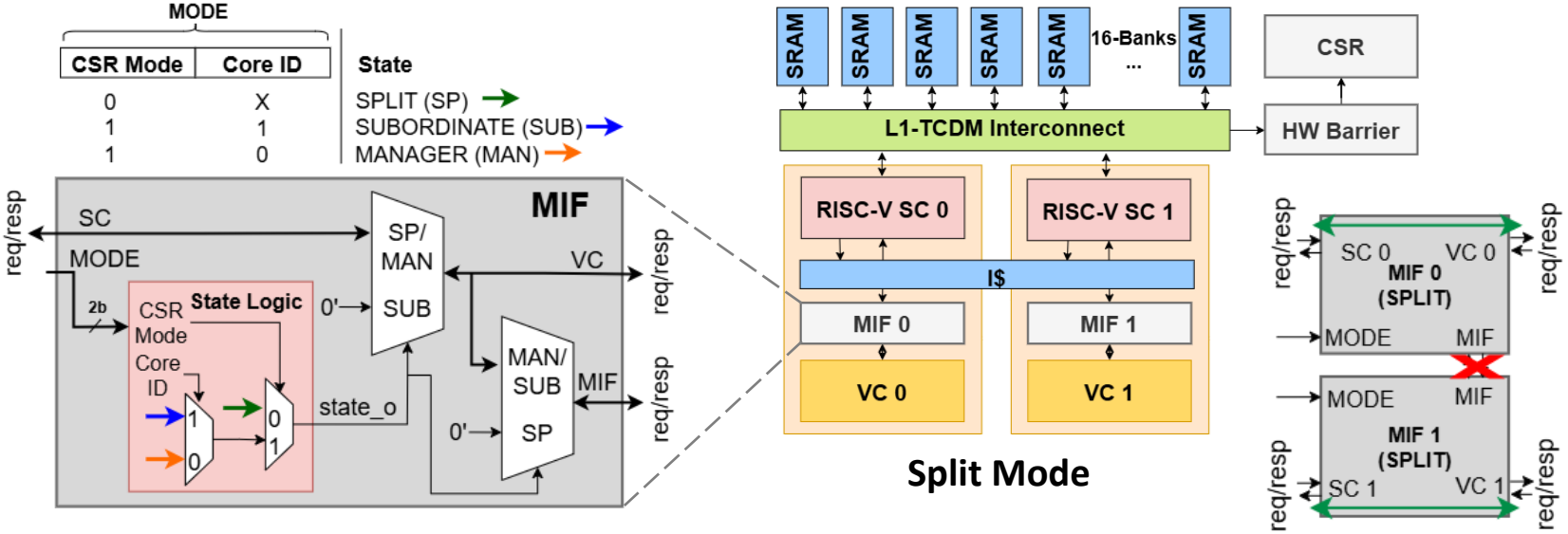



```
int main() {
  ...
  // Hardware barrier
  // wait for all cores to finish
  snrt_cluster_hw_barrier();
  // Hardware barrier
  // wait all cores to activate merge-mode
  snrt_cluster_hw_merge_mode();

  if (cid == 0) {
    // Data-parallel Vector workload
    ...
  }
  if (cid == 1) {
    // Pure scalar workload
    ...
  }
  // Hardware barrier
  // wait for all cores to finish
  snrt_cluster_hw_barrier();

  // Hardware barrier
  // wait all cores to activate split-mode
  snrt_cluster_hw_merge_mode();
  return 0;
}
```

The Reconfigurability Feature

Reconfiguration & Merge Interface

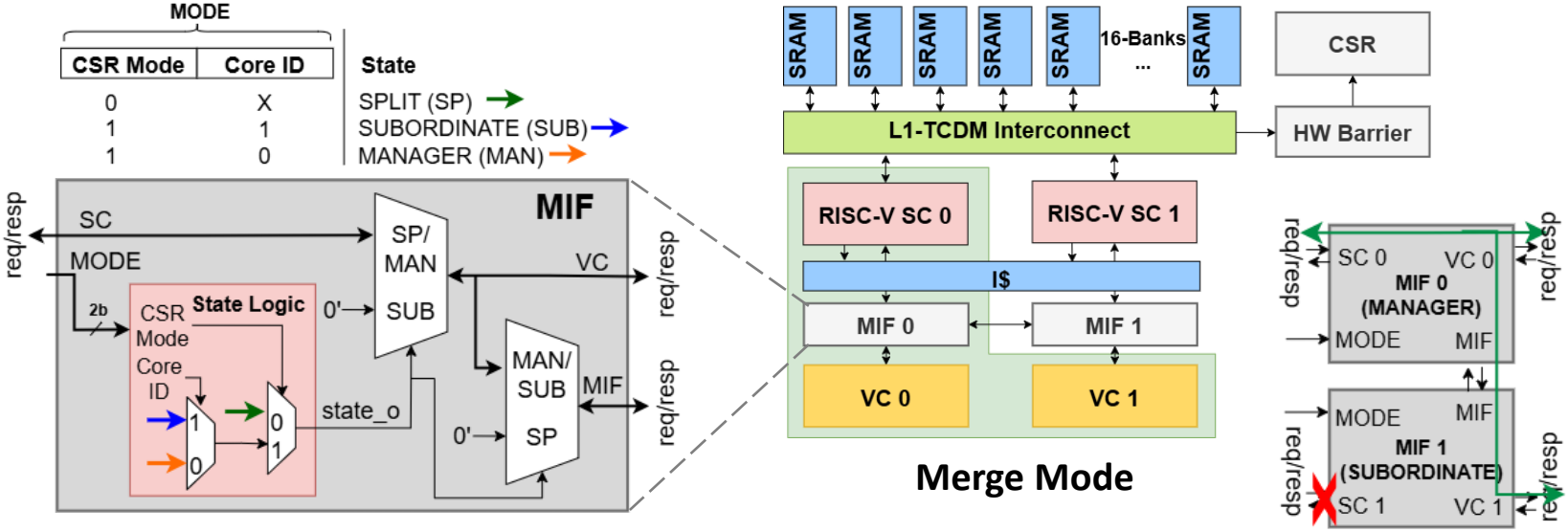
Each **fully combinational** MIF module forwards the instructions to the proper units based on the *Core ID* and the CSR *mode* register.



The Reconfigurability Feature

Reconfiguration & Merge Interface

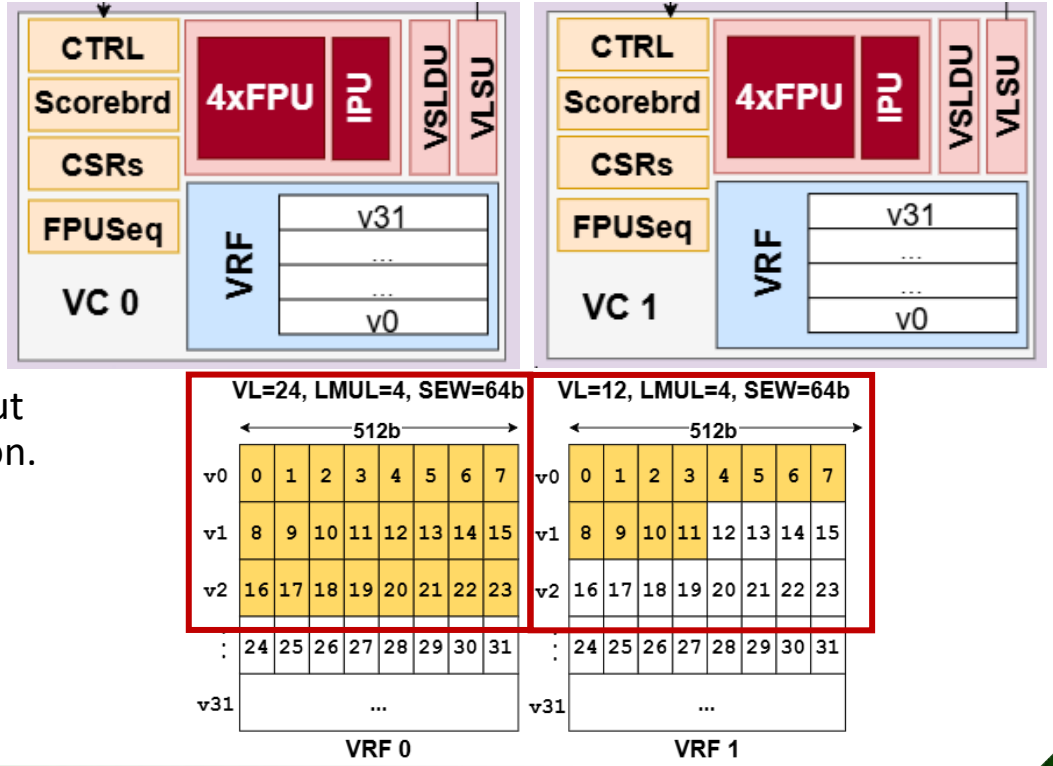
Each **fully combinational** MIF module forwards the instructions to the proper units based on the *Core ID* and the CSR *mode* register.



The Reconfigurability Feature

Vector Register File Layout – Split Mode

- VLEN of 512b.
- VRF organized in 4×64b wide banks
  - Each 3×64b read an 1x64b write ports (3R1W) → 3×256b/cycle read and 256b/cycle write bandwidth.
- The VRF's elements follows the byte-layout mapping defined by the RISC-V V extension.



The Reconfigurability Feature

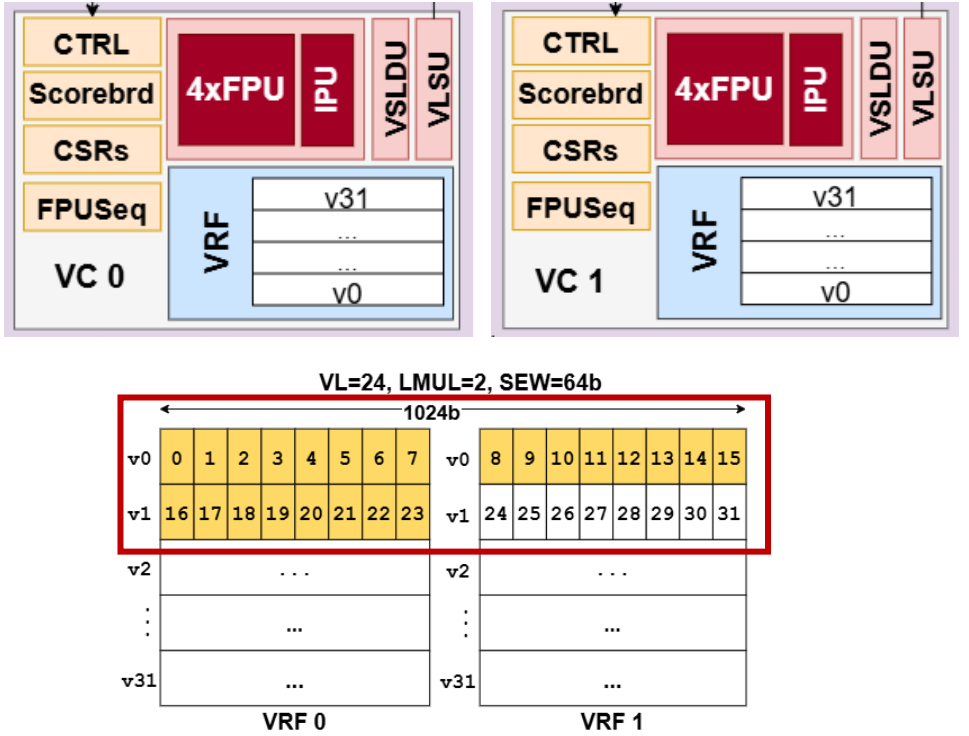
Vector Register File Layout – Merge Mode

- Unified VCs emulate a wide VRF with VLEN of 1024b.
  - 8×64b wide banks VRF each with 3R1W write port.
- The *ith* vector register of VC 0 and VC 1 are mapped respectively to the lower 0-3 and upper 4-7 banks.

✓ No Data reshuffling when SEW or LMUL change.

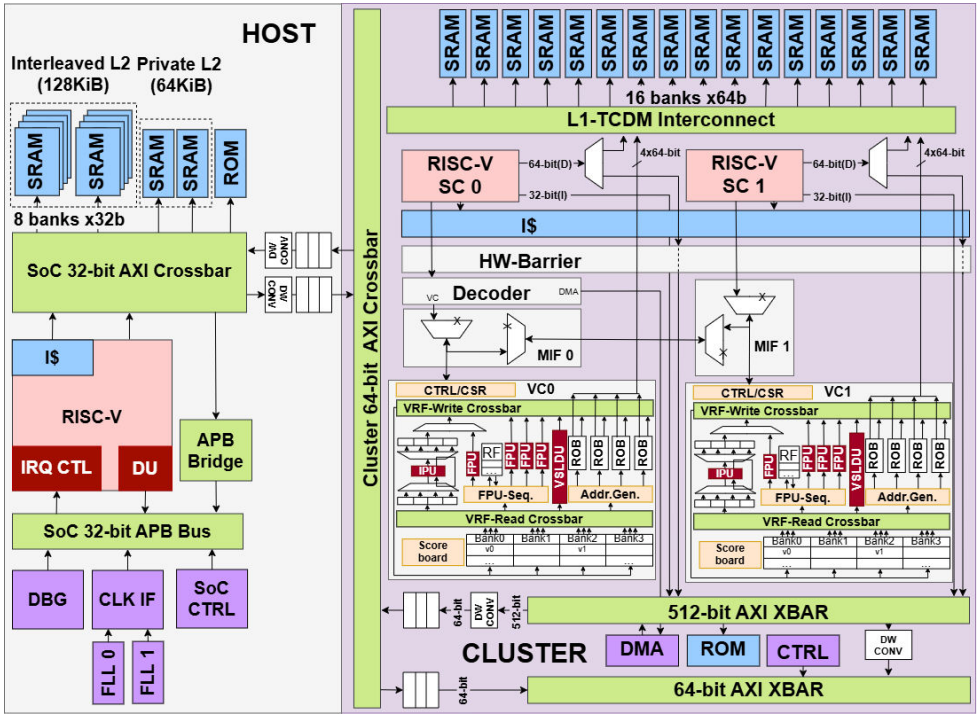
✓ Reuse of existing VRF R/W ports.

✓ The location of each operand remains predictable → For mixed-width operations each VC access operands from both VRFs.



The Buckbeak SoC

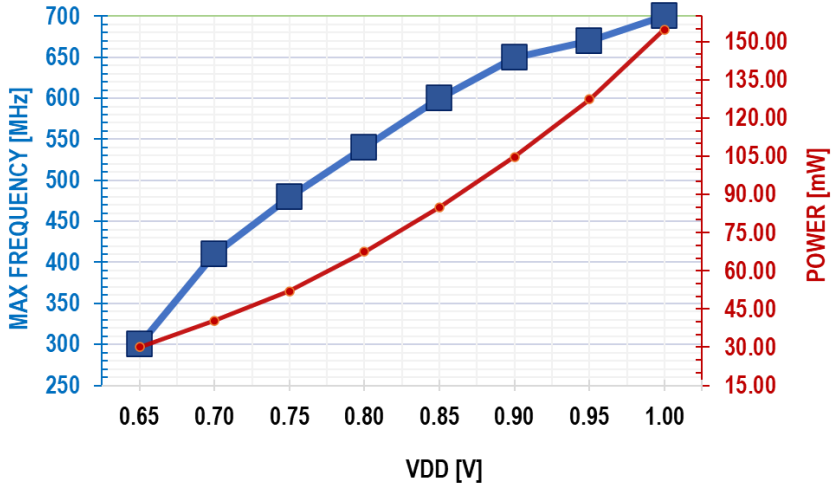
- **Host Domain**
  - RISC-V 32-bit RV32IMC Core
  - 128KiB L2 Shared SRAM
  - 64KiB L2 Private SRAM
  - Peripherals (Debug – FLLs – SoC CTRL)
- **Cluster Domain**
  - 2×RISC-V 32-bit RV32IMAFD Cores
  - 2×MIFs
  - 2×RVV1.0 Zve64d-based Vector Cores
    - 512b Vector Register File (VRF)
    - 4×64b Floating Point Unit (FPUs)
    - Vector Load/Store Unit (VLSU)
    - Vector Slide Unit (VSLDU)
    - Integer Processing Unit (IPU)
  - 128KiB L1 Shared SRAM
  - 512b DMA

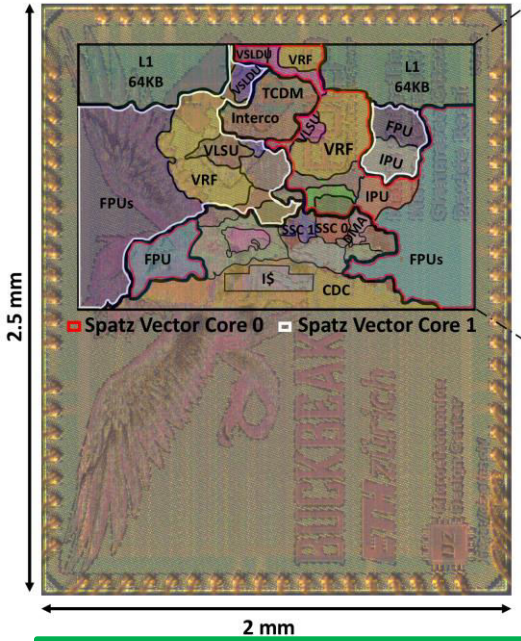


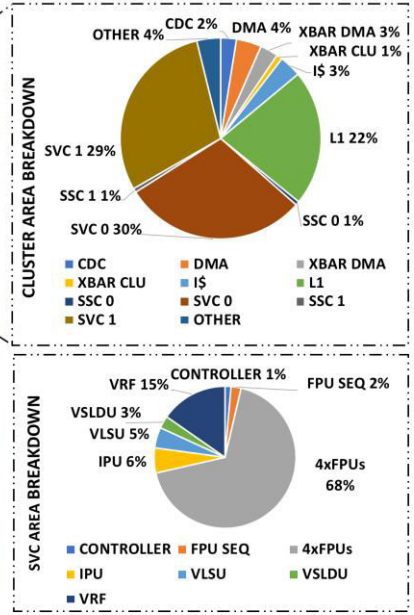
The Buckbeak SoC

Physical Implementation

Technology	Area	SRAM
GF 22nm FDX	5mm^2/1.5mm^2	128KiB
Supply	**Max Frequency**	**Power**
0.65V – 1V	300MHz – 700MHz	26mW – 155mW







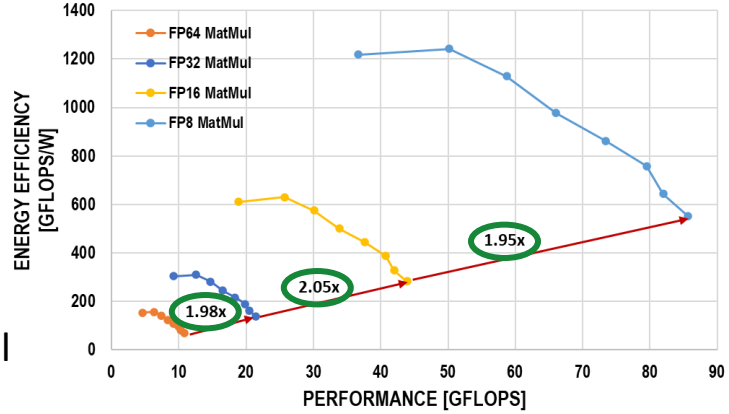
Reconfigurability logic takes only 0.04%!

The Buckbeak SoC

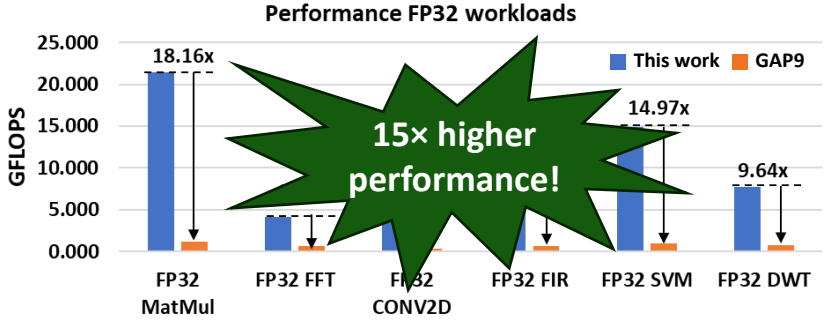
Performance and Energy Efficiency Results – Merge Mode

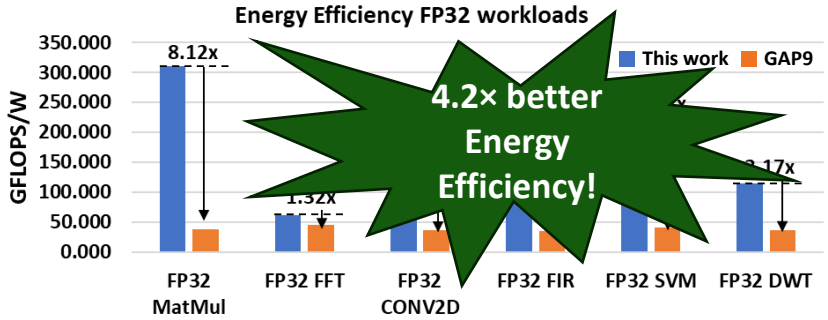
- **Precision Scaling**

Precision	Peak Performance	Peak Efficiency
FP8	85.64 GFLOPS	1.2 TFLOPS/W
FP16	43.89 GFLOPS	630 GFLOPS/W
FP32	21.40 GFLOPS	310 GFLOPS/W
FP64	10.78 GFLOPS	156 GFLOPS/W



- **FP32 near-sensor workloads** against GAP9[4], a commercial multi-core SoC in GF22 (0.65V@240MHz - 0.8V@370MHz)



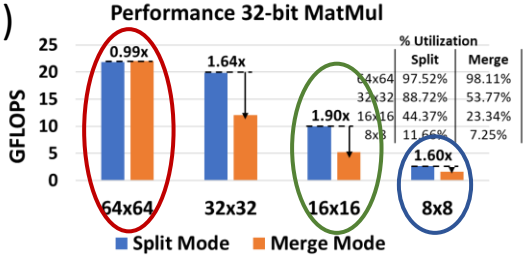


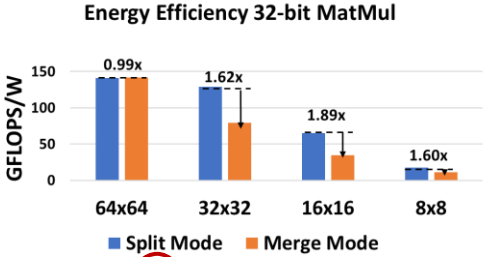
[4] D. Rossi et al., "Vega: A ten-core SoC for IoT endnodes with DNN acceleration and cognitive wake-up from MRAM-based state-retentive sleep mode," IEEE Journal of Solid-State Circuits, vol. 57, no. 1, pp. 127–139, 2022.

The Buckbeak SoC

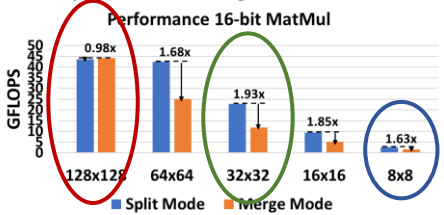
Impact of SoC reconfigurability – Workload scaling

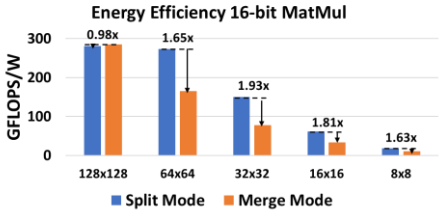
- Performance/Efficiency evaluation of Cluster adaptability of FP32 MatMul kernel on varying size at the best operating point (1V – 700MHz)
  - Split-Mode runs two kernels concurrently
  - Merge-Mode runs two kernels in succession on wider vector unit





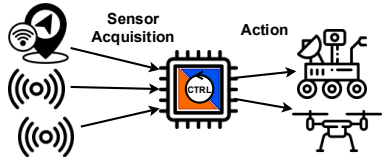
- **Large Matrices**
  Merge-Mode benefits from larger VR.
- **Medium Matrices**
  Split-Mode delivers the highest imprvement over Merge Mode.
- **Small Matrices**
  Split-Mode continues to deliver higher performance and efficiency, even as its speedup decreases.





The Buckbeak SoC

Impact of SoC reconfigurability – Mixed Scalar-Vector Workload



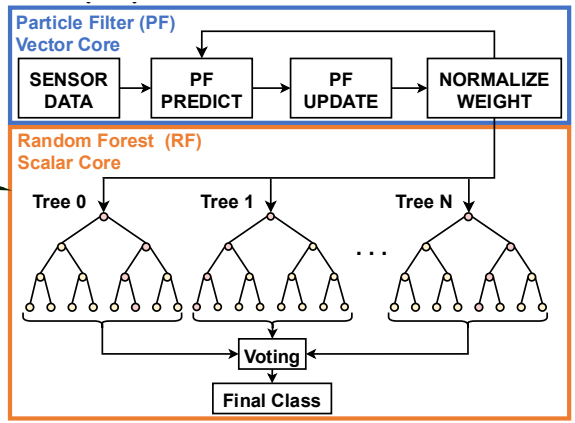
Mixed scalar–vector workload [5] profiling based on a real-time sensor fusion processing for nano-drone/rover position and velocity estimation.

- 2048-Particle Filter (PF) based on Sequential Monte Carlo (SMC) to estimate pose and velocity

  This profiling fuses 3 Sensors:
  - Global Navigation Satellite System
  - Wheel-odometry
  - Inertial Measurement Unit IMU yaw-rate

- Six-tree Random Forest (RF) to classifies and generate control actions

Vectorizable!



Scalar!

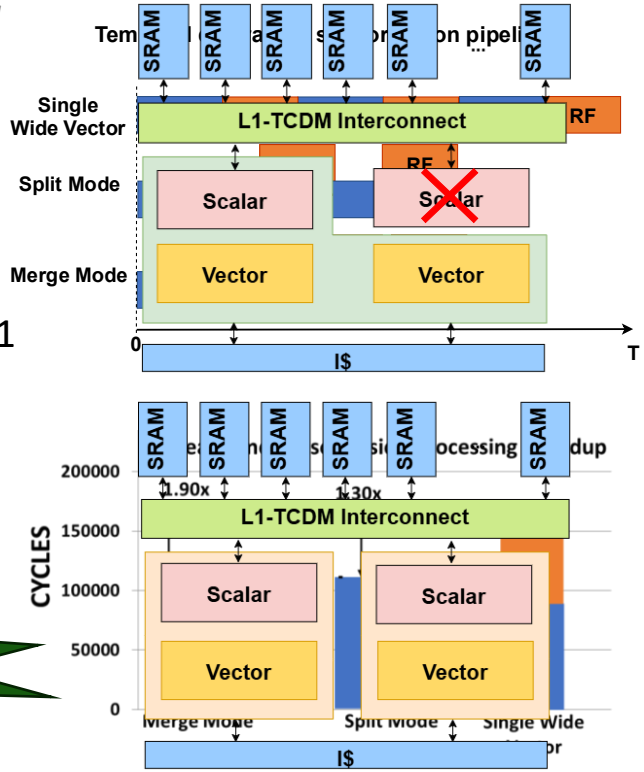
[5] N. Bonnor, "Principles of GNSS, inertial, and multisensor integrated navigation systems – second editionpaul d. groves artech house, 2013, 776 pp isbn-13: 978-1-60807-005-3," Journal of Navigation, vol. 67, no. 1, p. 191–192, 2014.

The Buckbeak SoC

Impact of SoC reconfigurability – Mixed Scalar-Vector Workload

The evaluation targets three configurations:

- **Single Wide Vector**
  A Single wide Vector Core (VC) is configured to match the effective VLEN of the merged unit (1024b).
- **Split Mode**
  Scalar Core (SC) 0 controls its local narrow (512b) VC and SC 1 executes scalar tasks.
- **Merge Mode**
  Both VCs operate as a single wider vector processor while freeing up SC 1 for executing non vectorizable kernels.



1.46× faster vs. Split Mode!

1.9× faster vs. Single Wide!

SoA Comparison

- **Low power (26–155 mW):** up to 51× higher peak performance and 24–74× better energy efficiency vs IoT MCUs used in nano-drones/rovers [6], [7].
- **Vector ISA reduces instruction-fetch overhead:** 3–4× better energy efficiency and 8–13× better area efficiency vs same-node programmable multi-core SoCs [4], [8].
- **Embedded vector**: 3–14.5× better energy efficiency and up to 15× better area efficiency vs HPC-oriented vector machines [9], [10].
- Compared to SoA CIM-capable embedded vector processor: 1.7× higher peak performance + wider precision support [11].

	Pixhawk 5 [6]	Leo Rover [7]	Marsellus [8]	VEGA [4]	[9]	YUN [10]	VECIM [11]	**(This work)**
Technology	90-nm	16-nm	22-nm CMOS FDX	22-nm CMOS FDSOI	Intel 16-nm FinFet	65-nm TSMC	65-nm TSMC	22-nm GF FDX
Die size	-	-	18.7 mm²	12mm²	24.01 mm²	6 mm²	4 mm²	5 mm²
Application	Drone + Rover	Small Rover	IoT GP + DNN + AI-IoT	IoT GP + NSAA + DNN	GP Vector SoC + DNN	GP Vector SoC	Computing in Memory	IoT GP
Architecture	Single Core	Multi Core	Multi Core	Multi Core	Vector	Vector	Vector CIM	Vector
Cores	STM32F765 + STM32F100 for I/O (Cortex M7)	BCM2712 (4x Cortex-A76)	16 x RV32IMCFXpulpnn + RV32IMCFXpulp + RBE	10 x RI5CY RVC32IMF-Xpulp	8 x RV64GC Xhwacha4 + 1 x RV64IMAC	RV64GCV0.9	RV64GCV0.9	1 x RV32IMC + 2 x RVV 1.0 ZVE64d
Supply Voltage	1.7V-3.6V	-	0.5V-0.8V	0.5V-0.8V	0.55V-1V	0.85V-1.2V	1V	0.65V-1V
Frequency	216MHz	2.4GHz	100MHz-420MHz	32kHz-450MHz	339MHz-1.44GHz	100MHz-280MHz	250MHz	300MHz-700MHz
Power range	100mW-150mW	1.5mW-12W	12.8mW-123mW	**1.7uW-49.4mW**	550mW-4.1W	57mW-330mW	-	26mW-155mW
Precision	FP64	FP64	INT2, INT4, INT8, INT16, INT32 RBE: 2-8 BF16, FP16, FP32	BFLOAT, FP16, FP32	FP16, FP32, FP64	FP64, FP32	INT8, FP16	**FP64, FP32, FP16 BF16, FP8 (4,3), FP8 (5,2)**
FPUs	1 x FP64	4 x FP64	8 x FP32	4 x FP32	64 x FP64	4 x FP64	16xFP16	8 x FP64
Peak Performance (GFLOPS)	0.21 (FP64)	9.6 (FP64)	6.9 (FP16)	3.3 (FP16) 2 (FP32)	**368.4 (FP16)**	2.83 (FP64) 5.70 (FP32)	25.3 (FP16)	**10.78 (FP64) 21.40 (FP32)** 43.89 (FP16) **85.64 (FP8)**
Peak Efficiency (GFLOPS/W)	2.1 (FP64)	6.4 (FP64)	207 (FP16)	129 (FP16) 79 (FP32)	56.5 (FP64) 92.3 (FP32) 209.5 (FP16)	10.8 (FP64) 23.4 (FP32)	230.1 (FP16)	**156.24 (FP64) 310.4 (FP32) 630 (FP16) 1242 (FP8)**
Area Efficiency (GFLOPS/mm²)	-	-	3.6 (FP16)	2.22 (FP16) 1.35 (FP32)	15.24 (FP16)	0.5 (FP64) 0.95 (FP32)	6.3 (FP16)	**7.2 (FP64) 14.3 (FP32) 29.3 (FP16) 57.1 (FP8)**

[4] D. Rossi et al., "Vega: A ten-core SoC for IoT endnodes with DNN acceleration and cognitive wake-up from MRAM-based state-retentive sleep mode," IEEE Journal of Solid-State Circuits, vol. 57, no. 1, pp. 127–139, 2022.
[6] A. S. M. A. S. Akib et al., "Efficient route planning and navigation in drones using pixhawk autopilot," in 2025 6th International Conference on Artificial Intelligence, Robotics and Control (AIRC), 2025, pp. 138–145.
[7] D. Ugur et al., "Fast and efficient terrain-aware motion planning for exploration rovers," in 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), 2021, pp. 1561–1567.
[8] F. Conti et al., "22.1 a 12.4TOPS/W @ 136GOPS ai-iot system-onchip with 16 RISC-V, 2-to-8b precision-scalable DNN acceleration and 30%-boost adaptive body biasing," in 2023 IEEE International Solid-State Circuits Conference (ISSCC), 2023, pp. 21–23.
[9] C. Schmidt et al., "An eight-core 1.44-GHz RISC-V vector processor in 16-nm FinFET," IEEE Journal of Solid-State Circuits, vol. 57, no. 1, pp. 140–152, 2022.
[10] M. Perotti et al., "Yun: An open-source, 64-bit RISC-V-based vector processor with multi-precision integer and floating-point support in 65-nm cmos," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 10, pp. 3732–3736, 2023.
[11] Y. Wang et al., "30.6 vecim: A 289.13GOPS/W RISC-V vector coprocessor with compute-in-memory vector register file for efficient high-performance computing," in 2024 IEEE International Solid-State Circuits Conference (ISSCC), vol. 67, 2024, pp. 492–494.

Conclusion

Silicon-proven heterogeneous RISC-V SoC for near-sensor computing featuring a 5-cycle reconfigurable dual-core vector cluster.

- Fabricated in GF 22nm FDX technology, it operates up to 700MHz at 1V within 155mW, achieving **up to 85.64GFLOPS** and **1.2TFLOPS/W (FP8).**
- Across FP32 near-sensor workloads, it outperforms a commercial multi-core SoC GAP9 **by 15× in performance** and **4.2× in energy efficiency**.
- In mixed scalar-vector workload, **Merge Mode** configuration **improves throughput by 1.9× over a Single wide** vector architecture **and 1.46× over Split Mode**.
- **Split Mode** operation **delivers 9.9GFLOPS and 65.27GFLOPS/W**, up to 1.9× over Merge Mode, showing superior performance & efficiency on FP32 mid-sized kernels.


OBRIGADO
gracias
どうも
ARIGATO
hvala
DANK U
謝謝
takk
MERSI
merci
obrigado
danke schön
köszi
سپاس
PALDIES
muchas gracias
ありがとう
TEŞEKKÜR EDERIM
MOLTE GRAZIE
GO RAIBH MAITH AGAT
danke
grazas
GRAZZI
THANKS
THANK YOU
благодаря
TAK
どうも
quan
TAK
asante
muchas gracias
vielen dank
PALDIES
OBRIGADO
DANK U
mesi
감사합니다
köszi
благодаря
grazie
TACK
DZIEKI
Gràcies
MULTUMESC
多謝
شكراً
danke
TEŞEKKÜR EDERIM
NA GODE
muchas gracias
obrigado
спасибо