

The European Processor Initiative

aka EPI

Question: what do they have in common?

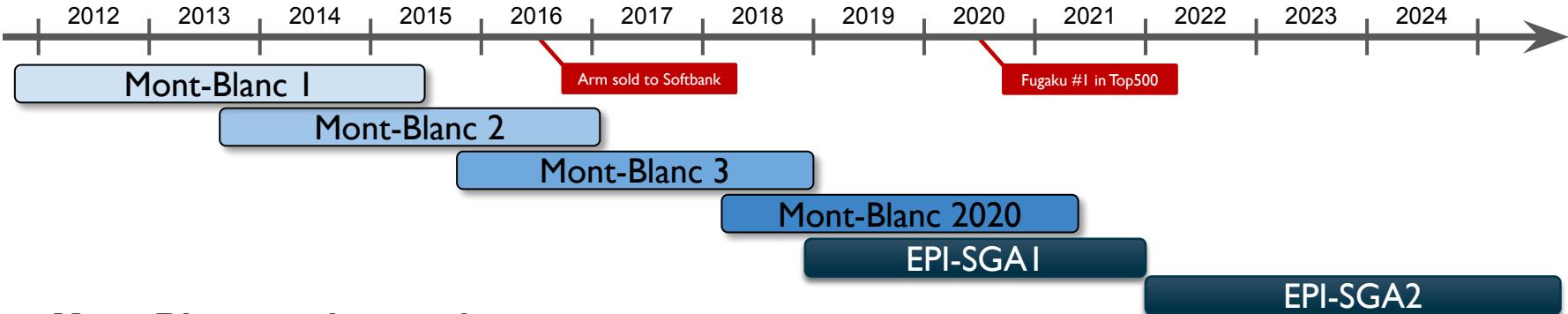


Apple iPhone 16



Fugaku supercomputer, 6th in the Top500 list

Europe looking for technological independence



- **Mont-Blanc project series**

- With the vision of leveraging the fast-growing market of mobile technology (aka Arm) for scientific computation, HPC and data centers
- Mont-Blanc 2020 first project where “hardware development” enters in the project plan

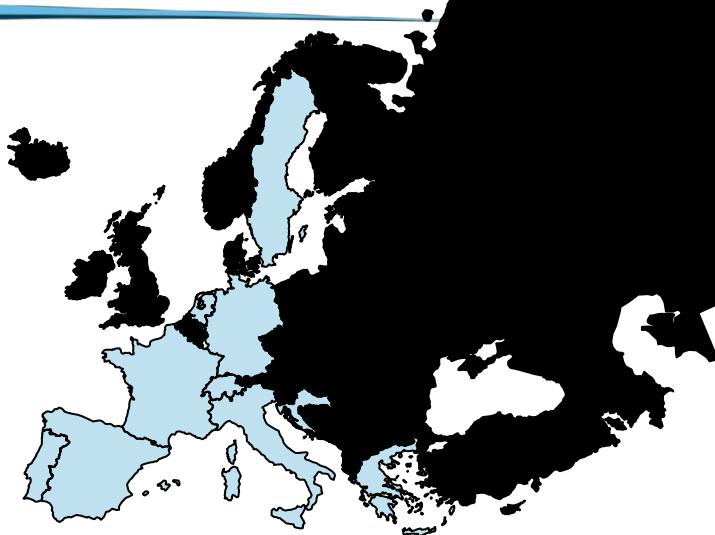
- **European Processor Initiative (EPI) series**

- No more “mobile”. Targeting HPC and automotive
- Fully focused on hardware development: General Purpose Processor (Arm) and Accelerator (RISC-V)



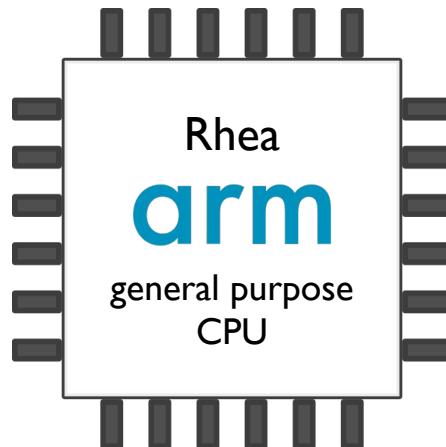
EPI project factsheet

- Phase 1 successfully concluded (2019-2021)
- Currently in Phase 2 (2022-2024)
- Consortium of 30 strategically chosen key European academic and industrial partners
- Funded by EuroHPC JU (50%) and co-funded by Croatia, France, Germany, Greece, Italy, the Netherlands, Portugal, Spain, Sweden and Switzerland
- Total budget: 70 M€

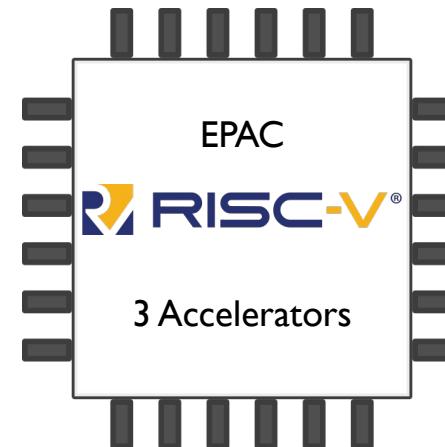


EPI Main Objective

- To develop European microprocessor and accelerator technology
- Strengthen competitiveness of EU industry and science



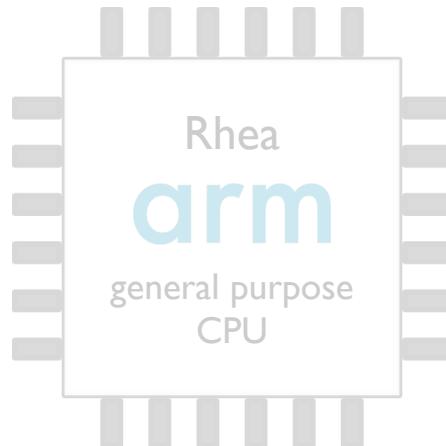
SiPearl, Atos, CEA, UniBo,
E4, UniPi, P&R



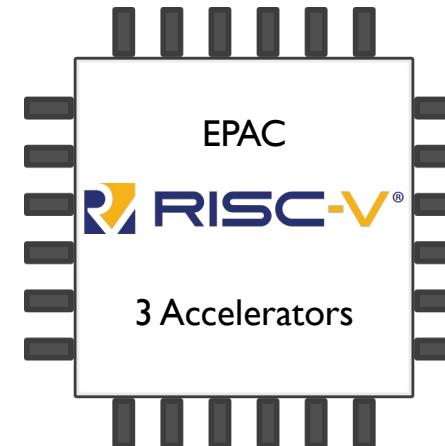
BSC, SemiDynamics, EXTOLL, FORTH,
ETHZ, UniBo, UniZG, Chalmers, CEA, E4

EPI Main Objective

- To develop European microprocessor and accelerator technology
- Strengthen competitiveness of EU industry and science



SiPearl, Atos, CEA, UniBo,
E4, UniPi, P&R



BSC, SemiDynamics, EXTOLL, FORTH,
ETHZ, UniBo, UniZG, Chalmers, CEA, E4

EPAC: EPI Accelerator v1.5

GF22FDX, 27 mm², 0.3 Btr Tape out Mar 2023, Bring up Oct 2023

VEC tile

General purpose RISC-V CPU
Avispado Core (16 kI\$, 32 kD\$)
with dedicated VPU
Up to 256 DP element vector length



VRP tile

General purpose RISC-V CPU
supporting variable precision
arithmetic up to 256 bit elements



L2-HN tile

Distributed L2 cache (256 kB/slice) and
Coherence Home Node



Physical design by Fraunhofer
Prototype board integration by E4
COMPUTER ENGINEERING

STX tile

RISC-V many-core machine learning
accelerator targeting stencil and
tensor arithmetics.



CHI NoC and SerDes

On-chip high-speed network based
on multiple CHI cross points (XP).

Off-chip link based on SerDes.



EPAC: EPI Accelerator v1.5

GF22FDX, 27 mm², 0.3 Btr Tape out Mar 2023, Bring up Oct 2023

VEC tile

General purpose RISC-V CPU
Avispado Core (16 kI\$, 32 kD\$)
with dedicated VPU
Up to 256 DP element vector length



VRP tile

General purpose RISC-V CPU
supporting variable precision
arithmetic up to 256 bit elements



L2-HN tile

Distributed L2 cache (256 kB/slice) and
Coherence Home Node



Physical design by Fraunhofer
Prototype board integration by E4
COMPUTER
ENGINEERING

STX tile

RISC-V many-core machine learning
accelerator targeting stencil and
tensor arithmetics.



CHI NoC and SerDes

On-chip high-speed network based
on multiple CHI cross points (XP).

Off-chip link based on SerDes.



What's special in EPAC – VEC?

The “Avispado” RISC-V core



The Vector Processing Unit (VPU)



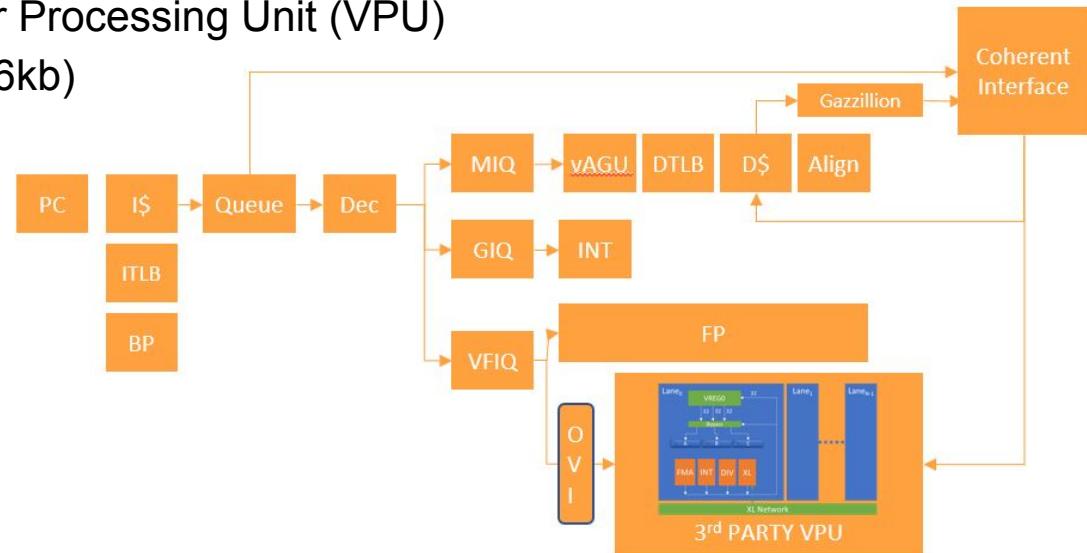
**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

What's special?

- It boots Linux
- The scalar in-order RISC-V core releases several requests of cache lines to the memory
- The core is connected to a Vector Processing Unit (VPU) with very wide vector registers (16kb)

- 16 kB instruction cache
- 32 kB L1 data cache
- 1 MB L2 cache
- Decodes RVV v0.7 vector extension
- Cache coherent (CHI)
- Vector memory accesses (vle, vlse, vlxe, vse, ...) processed by a dedicated queue (MIQ/LSU)



Courtesy:

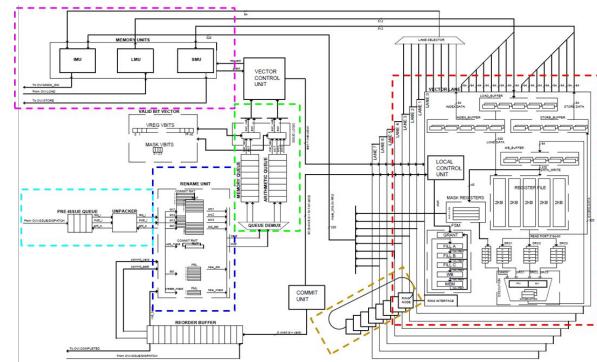
semidynamics
silicon design and verification services

EPAC-VEC: Vector processing unit “Vitruvius”

Architecture	Vector register size (1 cell = 1 double element)																...	D256
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	...	D256
Intel AVX512	D1	D2	D3	D4	D5	D6	D7	D8										
Arm Neon	D1	D2																
Arm SVE @ A64FX	D1	D2	D3	D4	D5	D6	D7	D8										
NEC Aurora SX	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	...	D256
RISC-V EPAC Vec	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	...	D256

Implementation

- Long vectors: 256 DP elements
 - #Functional Units (FUs) << Vector Length (VL)
 - 1 vector instruction can take several (32) cycles
- 8 Lanes per core
 - FMA/lane: 2 DP Flop/cycle
- 40 physical registers, some out of order
- Vector length agnostic (VLA) programming and architecture



F. Minervini, et al. “Vitruvius+: An Area-Efficient RISC-V Decoupled Vector Coprocessor for High Performance Computing Applications” [TACO-2022-50]

VPU with Long Vector Length (VL) support



— AVX512



512 bits per vector (8 DP elements)

arm — SVE



Up to 2048 bits per vector (32 DP elements)

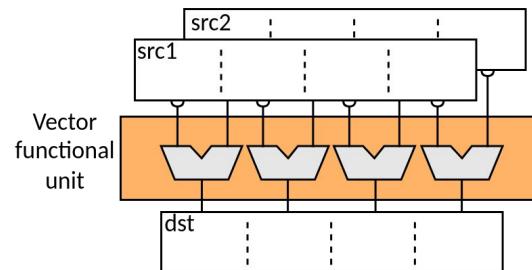
NEC / RISC-V



16384 bits per vector
(256 DP elements)

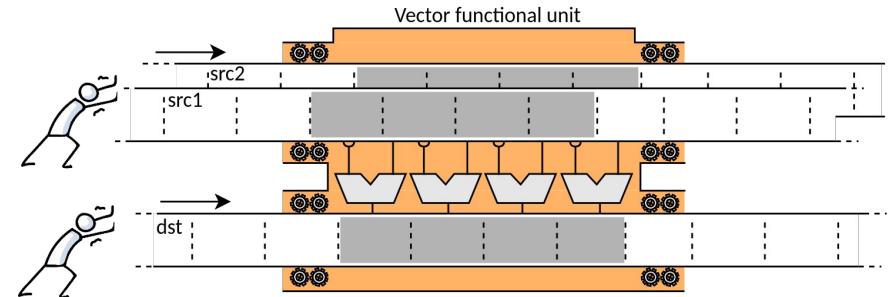
Short VL

- As many Functional Units as VL.
- Vector instructions executed in 1 cycle

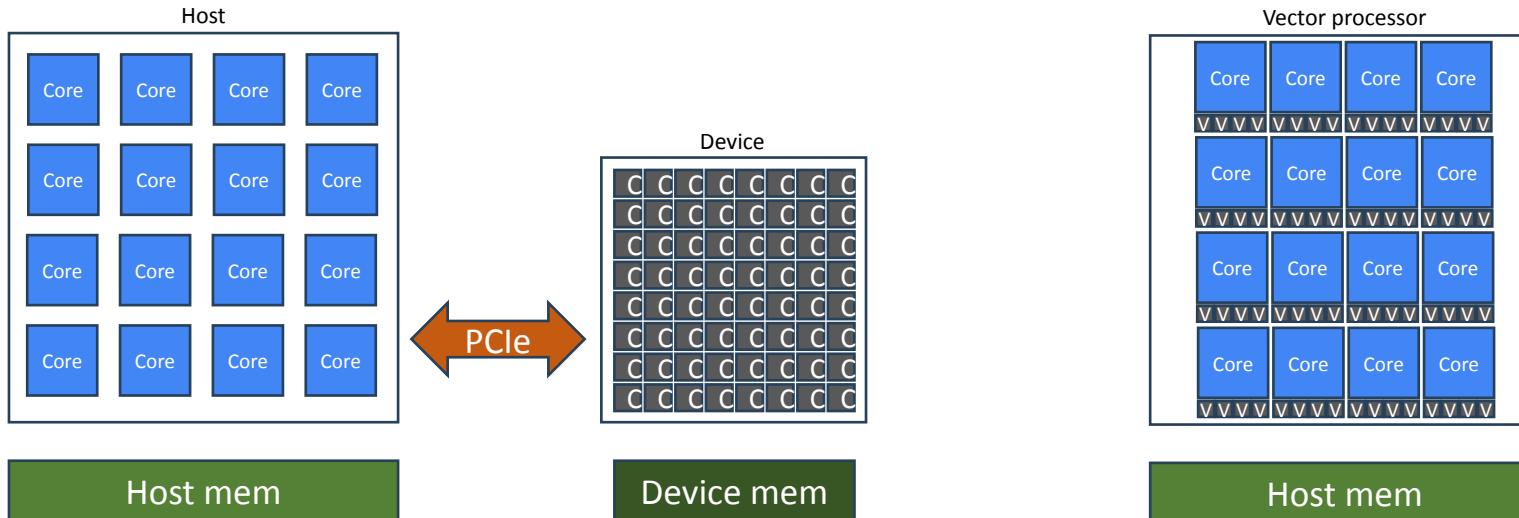


Long VL

- Cannot afford (area, power, cost) hundreds of Functional Units
- Vector instructions are executed on multiple cycles



Host-device vs Vector

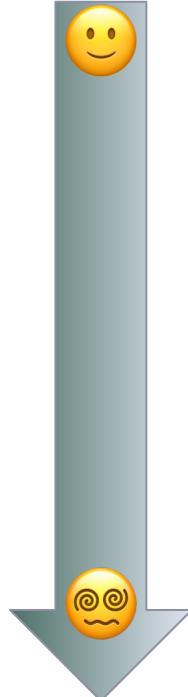


Sizing an HPC system

What if we want to make EPAC as powerful as (half) of a top-end GPU?

	EPAC @ 1GHz (1 core)	NVIDIA H100
Performance	16 GFLOP/s * x	30 TFLOP/s
Area [mm²]	0.1 Btr * y 10 mm ² * z	? Btr 830 mm ²
Cost	Not for sell 😎	30 k€

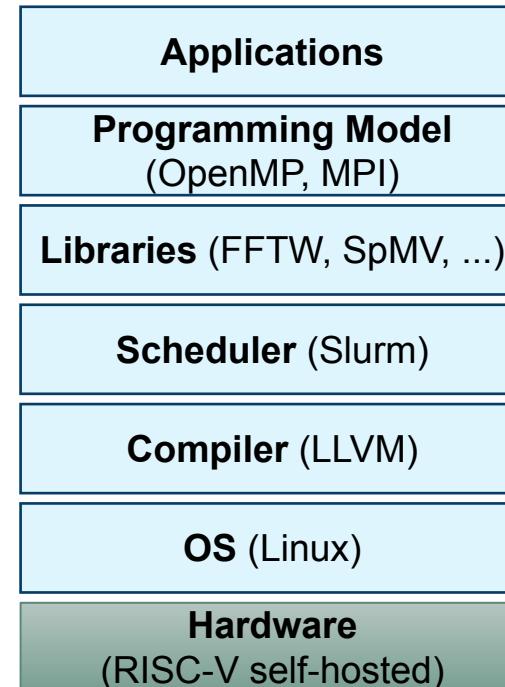
How do I program EPAC – VEC?



- **Autovectorization**
 - Leave it to the compiler
- **#pragma omp simd** (aka “Guided vectorization”)
 - Relies on vectorization capabilities of the compiler
 - Usually works but gets complicated if the code calls functions
 - Also usable in Fortran
- **C/C++/FORTRAN builtins** (aka “Intrinsics”)
 - Low-level mapping to the instructions
 - Allows embedding it into an existing C/C++ codebase
 - Allows relatively quick experimentation
- **Assembler**
 - Always a valid option but not the most pleasant

How do I use EPAC - VEC?

- Like a standard HPC system!
- Compile your code
 - We give you a compiler
- Link libraries
- Write/Submit a job script
 - SLURM
- Wait for the results
- Analyse execution traces and study how well your code is vectorized

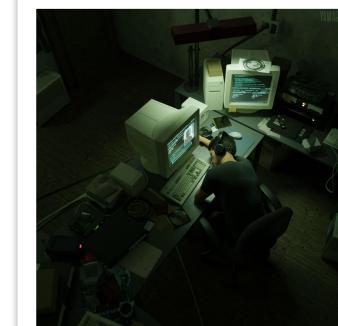
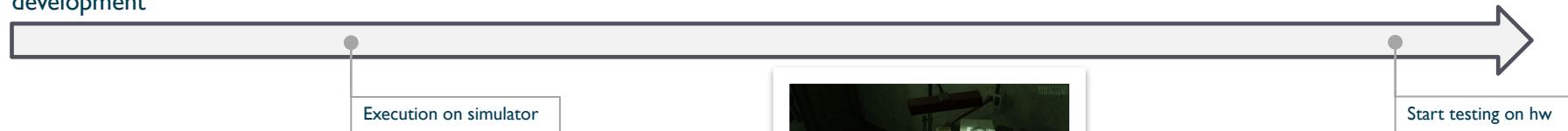


What to do until the hardware is ready?

Hardware development



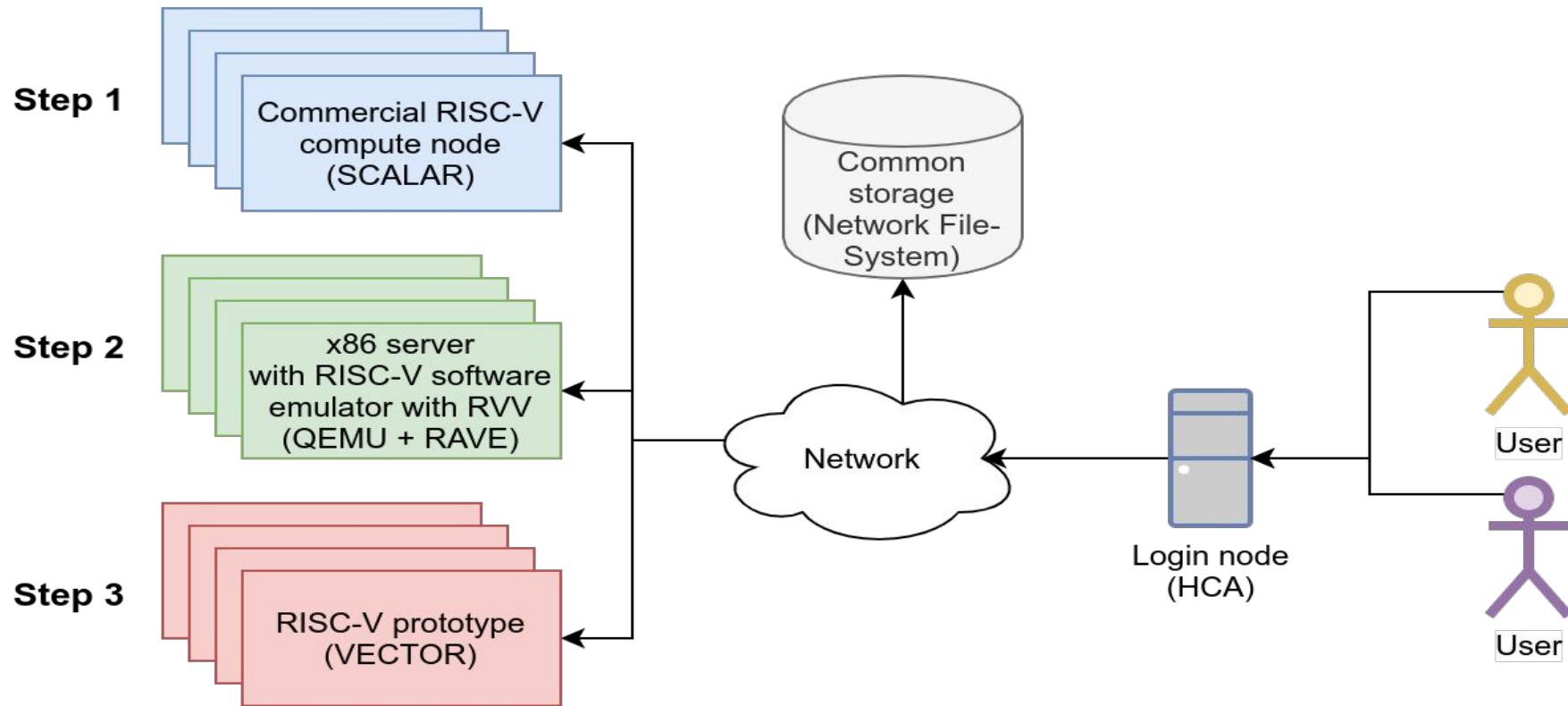
Software development



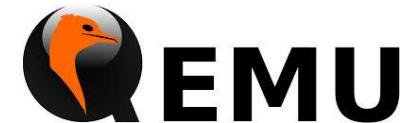
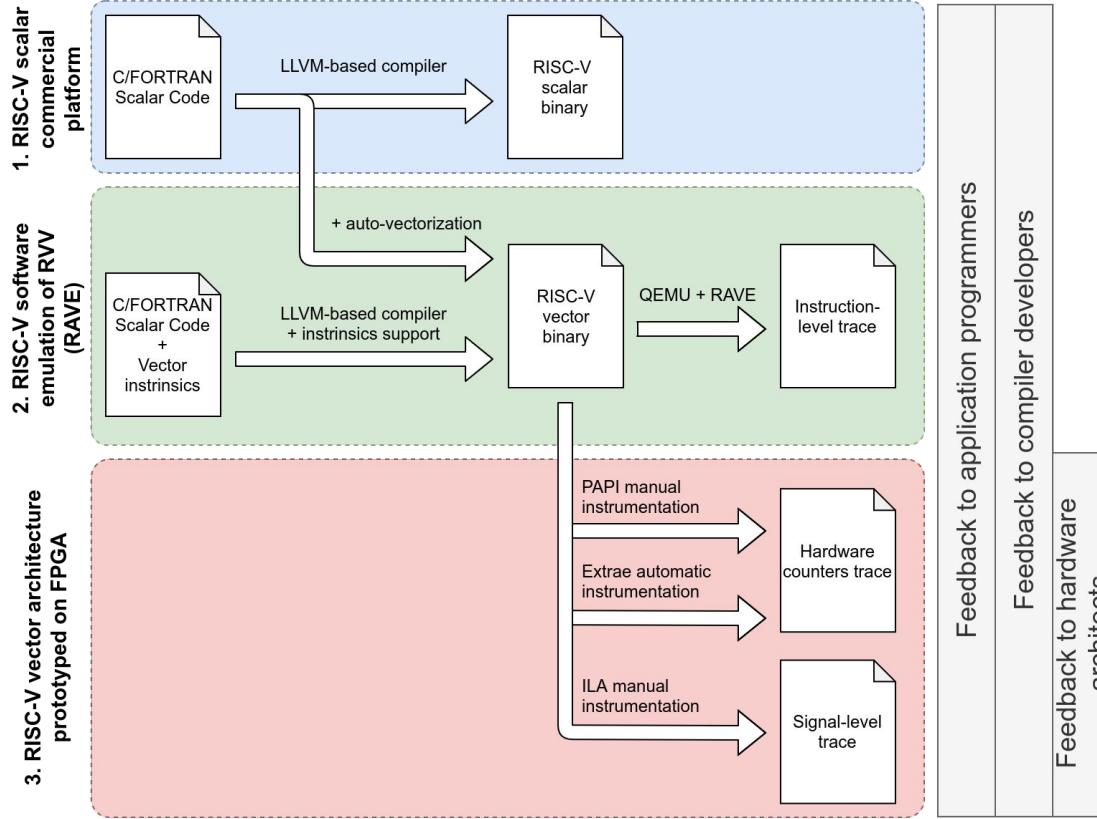
Wake up Neo... Follow the Software Development Vehicles



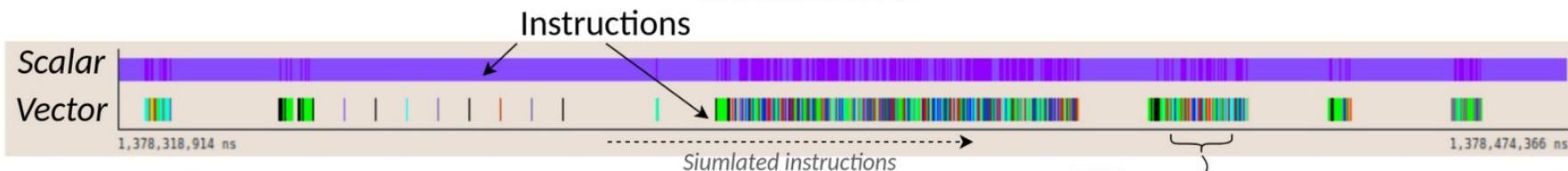
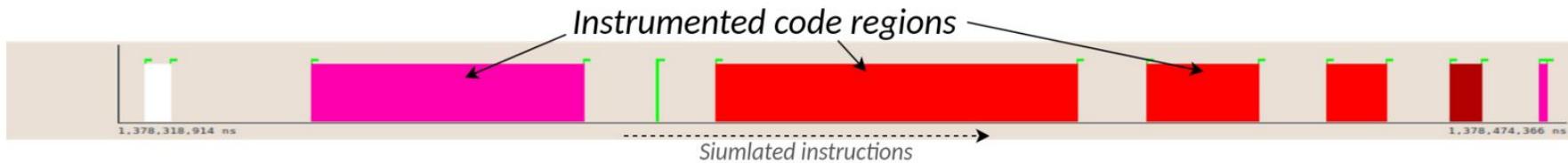
Software Development Vehicles (SDV)



Co-design with SDV

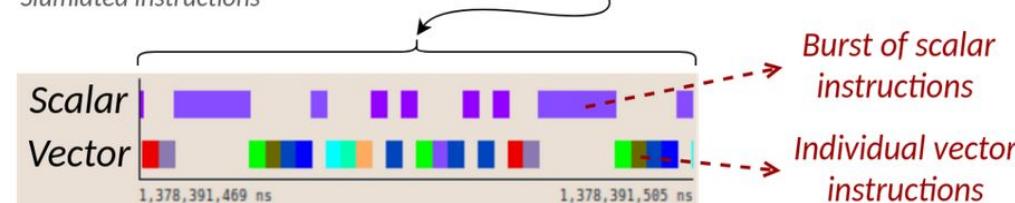


Navigate, visualize and quantify



Prop. of Scalar and Vector instructions

	Init	BU	TD	Bit->Q (c)	Q->Bit (e)
qemu_scalar	0.62	0.63	1.00	0.81	0.74
qemu_vector	0.38	0.37	0.00	0.19	0.26



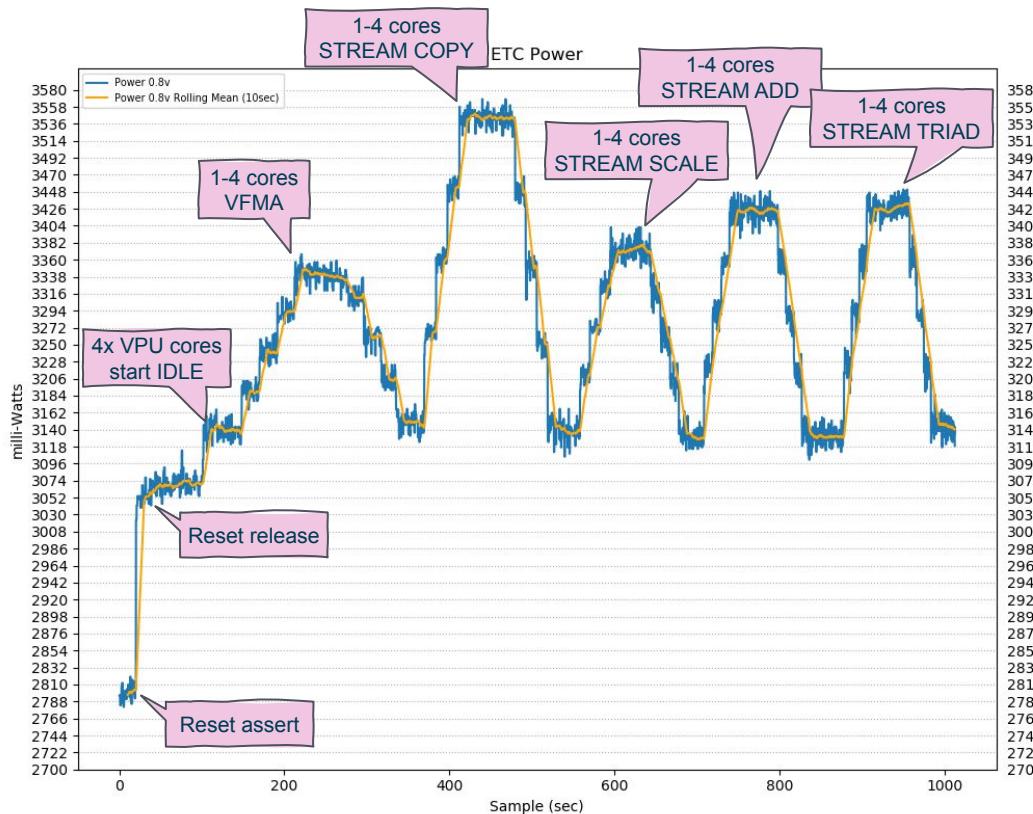
Assembly of instructions (In phase Init)

	vsub	vsll	vsetvl	vle	vmerge	vmv	vsxe	vmseq	scalar
qemu_scalar	-	-	640	-	-	-	-	-	641
qemu_vector	128	128	-	256	128	128	128	128	-

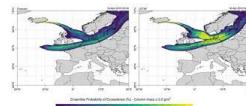
Average bits per instruction (In phase Init)

vsub	vsll	vle	vmerge	vmv	vsxe	vmseq
16,382.50	16,382.50	16,382.50	16,384	16,384	16,382.50	16,382.50

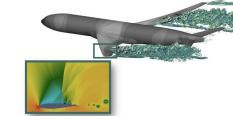
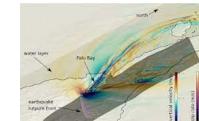
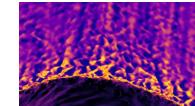
Not only performance...



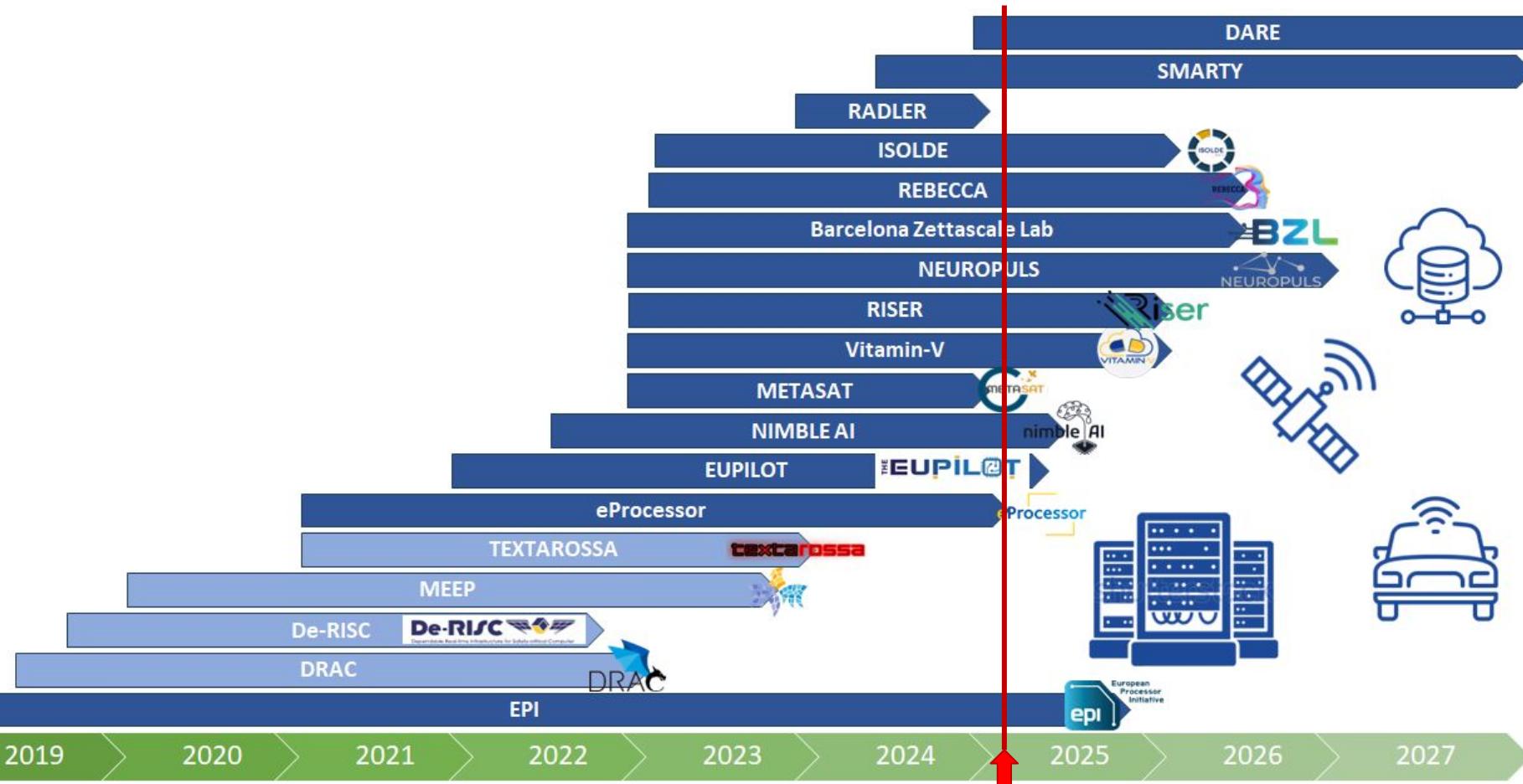
Real science



Fall3D



What comes next?



Digital Autonomy with RISC-V in Europe: DARE

Objectives

- to build prototype HPC and AI systems based on technology designed and developed in Europe
- to leverage chiplet technology
- to deliver prototypes of GPP, AIPU, and VEC on latest silicon technology
- to develop the system software for facilitating the adoptions of DARE platforms

Partners:

- Codasip (GPP), Axelera AI (AIPU), OpenChip (VEC), IMEC, JSC, BSC
- 38+7 partners among industry, datacenter and academia in Europe

Time and budget

- 3 years + 3 years
- 240 M€ for 2025-2027

Take-home message

- EPI is developing:
 - Arm-based CPU
 - RISC-V-based Accelerator
- We focus on the RISC-V vector accelerator (VEC) that:
 - Can be self-hosted
 - Support variable vector length
 - Is vector length agnostic
 - Uses long vectors (256 DP elements, 32x larger than x86)
- While chips are becoming available, EPI develops tools for boosting the co-design cycle
 - Software and Hardware prototypes (aka Software Development Vehicles)
- We can leverage SDVs to:
 - Influence hardware design
 - Improve compiler autovectorization and system-software support
 - Study and improve vectorization of real scientific HPC codes
- DARE is the next macro-project pushing RISC-V in Europe



Centre of Excellence in Exascale CFD



References

-  Mantovani, Filippo, et al. "Software Development Vehicles to enable extended and early co-design: a RISC-V and HPC case of study." International Conference on High Performance Computing. Cham: Springer Nature Switzerland, 2023. <https://arxiv.org/abs/2306.01797>
-  Vizcaino, Pablo, et al. "Short reasons for long vectors in HPC CPUs: a study based on RISC-V." Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis. 2023. <https://arxiv.org/abs/2309.06865>
-  Vizcaino, Pablo, et al. "RAVE: RISC-V Analyzer of Vector Executions, a QEMU tracing plugin." arXiv preprint arXiv:2409.13639 (2024). <https://arxiv.org/abs/2409.13639>
-  Blancafort, Marc, et al. "Exploiting long vectors with a CFD code: a co-design show case." 2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2024. <https://arxiv.org/abs/2411.00815>

-  <https://www.eetimes.com/examining-the-top-five-fallacies-about-risc-v/>
-  <https://www.youtube.com/watch?v=iFlcJFcOJKk>



Google Scholar

EPI FUNDING



SPONSORED BY THE



This research has received funding from the European High Performance Computing Joint Undertaking (JU) under Framework Partnership Agreement No 800928 (European Processor Initiative) and Specific Grant Agreement No 101036168 (EPI SGA2). The JU receives support from the European Union's Horizon 2020 research and innovation programme and from Croatia, France, Germany, Greece, Italy, Netherlands, Portugal, Spain, Sweden, and Switzerland. The EPI-SGA2 project, PCI2022-132935 is also co-funded by MCIN/AEI /10.13039/501100011033 and by the UE NextGenerationEU/PRTR.



Swedish
Research
Council



Financé
par

