RISC-V®

State of the Union

Krste Asanovic
Chief Architect, RISC-V International

RISC-V Summit
Paris, France
May 13, 2025

The State of the Union is strong!

Solidly established in embedded space

On verge of widespread adoption in application processors

The standard base for new AI accelerators

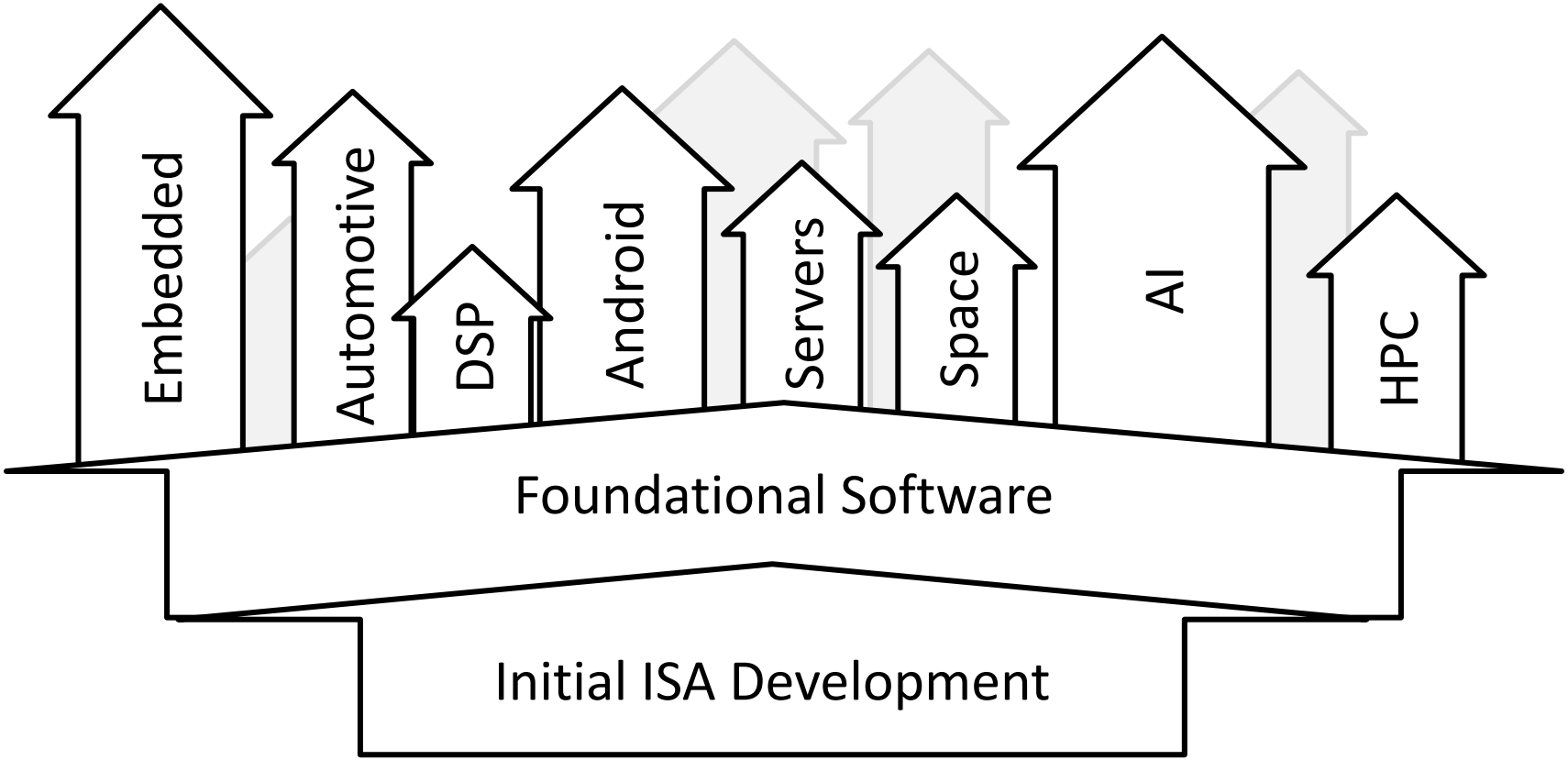
Increasing industry realization that open-standard RISC-V will be the dominant future ISA

RISC-V is 15 Years Old!

RISC-V Official Birthday is May 18, 2010

2010-2014 initial development at UCB and roll out to broader community
2015-2019 initial standardization, RV64GC, first commercial offerings
2020-2024 widespread commercial embedded deployments, filling out standards, RVA23, building foundational software
2025- what next?

RISC-V move into industry verticals



Attacking each Vertical

- Begin with common ISA and foundational software baselines
- Work with relevant software ecosystem partners
  - Work is done at RVIA in SIGs (members only) and Joint Working Groups (include non-member ecosystem partners)
- Porting and RISC-V tuning of key frameworks and libraries needed to ascertain gaps and opportunities
- Can only be successful by having complete solution for each vertical, so need to prioritize RVIA bandwidth
  - TSC and BoD gathering feedback from members
- Vertical efforts benefit from leveraging commonality across whole RISC-V universe, ensuring common evolution of components

Profile Progress

- RVI20, RVA20, RVA22, RVA23, RVB23 profiles ratified

- RVA23 was last major release in RVA profile family
- Starting to plan next minor release RVA23p1, which will only add options
- Expect a new RVA23 minor release, 1-2 per year, as options ratified
- Next RVA major release, tentatively named RVA30, which introduces new mandates not expected for another ~2-3 years

- Enthusiasm building to define a microcontroller profile RVM (draft RVM23 available) as well as an automotive MCU profile

Specification Improvements

Tremendous effort by staff and volunteers to pull together all ISA specifications into one unified document tree and improve:

https://github.com/riscv/riscv-isa-manual

- Goal is to render all specification content in different formats (human- and/or machine-readable) from this repo
  - Still much work to do to clean up and restructure, please help!
  - Work with documentation SIG, UDB SIG, and github repo to give corrections, feedback, and input on future document tree structure
- New version of ratified-only specifications published for this summit
  - RISC-V ISA Manual, version 20250508
  - Removes historical and/or non-ratified material from published spec

RISC-V New Security Extensions in Progress

- SPMP
  - Provide S-mode RTOS/supervisors with protection from U-mode tasks
- RISC-V Worlds
  - Provide secure global partitioning of devices on SoCs
- Supervisor Domains (Smmtt)
  - Flexible support for confidential computing and other applications
- CHERI
  - New base ISAs (RV32Y/RV64Y) bringing capabilities to RISC-V
- Lightweight Memory Tagging
  - Improve memory safety, building upon pointer-masking support
- Additional Crypto
  - PQC, and other vector crypto enhancements

RISC-V DSP extensions

- RISC-V P extension
  - Long development time, but now stable on path to ratification
  - Packed integer/fixed-point (8b/16b/32b/64b) operations on integer **x** registers
  - Over 100 new instructions

- Upcoming RISC-V vector DSP extension
  - Add DSP instructions on **v** registers
  - Candidate new instructions for greater fixed-point support, permutations for FFTs, complex arithmetic support

Long (>32b) Instructions

- RISC-V ISA included variable-length instructions from beginning
- Compressed instructions (16b) save code-size
- RISC-V designed for long-term success, won't disappear due to owner changing business model or folding. Fixed 32b instruction format would be barrier to long-term evolution. Other fixed-width 32b ISAs already running out of encoding room.
- Longer instructions (>32b) also help reduce code size, improve performance, and support ever-growing number new operators and datatypes
- New TG being formed to finalize encoding format and first long instructions

AI Computing

AI is pervasive, both as:

- Horizontal technology
  - Used everywhere on every device
  - Single device might run dozens of different models
- Vertical application
  - Dedicated racks of machines for training models and serving inferences

- Common attributes:
  - Large amounts of numeric computation on small specialized datatypes
  - Large memory footprint to hold model parameters
  - Large memory bandwidth to fetch parameters and communicate between layers

AI Evolution

AI is in stage of rapid evolution with new model architectures, new algorithms, new operators, new datatypes. Hardware must be flexible, as AI software change rate much greater than hardware replacement rate.

Most current models read all weights on every inference, but newer models will selectively compute on subsets of weights

More complex control and scheduling will probably be needed for future more advanced models.

RISC-V and AI

- RISC-V supports a general computing model allowing a balance of scalar, vector, and matrix capabilities
  - Same cores can run non-AI portion of application as well as AI portion, simplifying software, lowering latency, and increasing utilization
  - Can scale to arbitrarily high performance with scale-up+scale-out multicores
  - New models can run reasonably well on RISC-V hardware designed for earlier models
  - Different balance of scalar/vector/matrix possible with different microarchitectures in current or future generations of RISC-V, without changing programming model

RISC-V AI Vector Datatypes

- Proposal (Zvfbfa) for additional support for BF16 vector arithmetic
- Proposal (Zvfofp8min) for OFP8 support through conversions

- More datatype activity likely

RISC-V Matrix Extensions

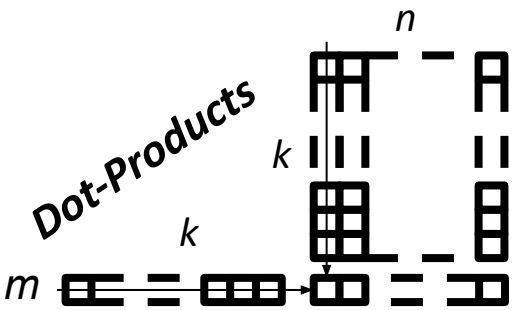
- Matrix multiply (matmul) instructions can provide large speedups on key compute in portions of AI applications
- Across universe of RISC-V implementations and application domains, different matmul microarchitectures make sense, and may result in different ISA choices (or no matmul at all)
- Only a relatively small number of matrix multiply routines exist, so software less affected by ISA choices

RISC-V Matrix Extensions, 1+3 Approaches

Element in existing RVV vector register

Element in added matrix registers

Vector Extension

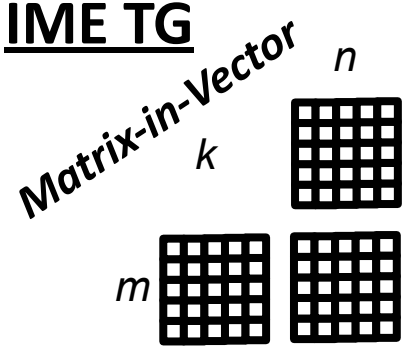


Vector Func. Unit

RVV Vectors

Memory

IME TG

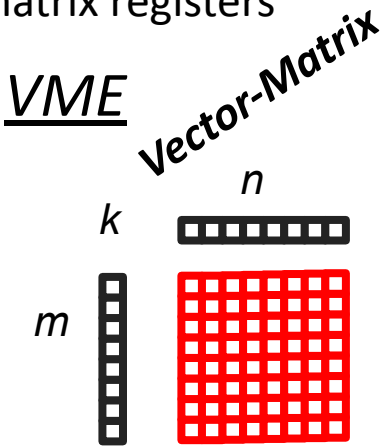


Matrix Func. Unit

RVV Vectors

Memory

VME

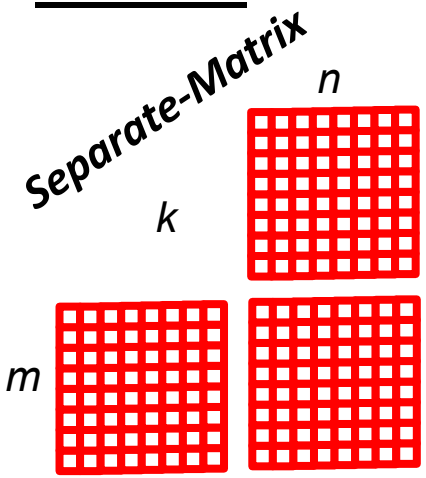


Matrix Engine

RVV Vectors

Memory

AME TG



Matrix/Vector Engine

Memory

Alternative Matrix Instruction Extensions

- **Dot-Products:** No additional matrix state. Small and effective extension at smaller vector lengths, also meets some DSP needs, but doesn't scale performance with larger vector lengths.
- **Matrix-in-Vector:** No additional matrix state. Allows larger throughputs than dot-products for longer vector lengths.
- **Vector-Matrix:** Add matrix state to processors that already have RVV. Maintains vector ISA and memory model.
- **Separate-Matrix:** Unconstrained matrix and vector design.

Summary

- RISC-V foundational components in place and working well
- RISC-V moving into verticals, each vertical will need focused effort to engage ecosystem and fill gaps in ISA or software support, while keeping coherent overall ISA design
- RISC-V well-suited to current and future AI needs. Matrix extensions possibly liveliest part of RISC-V standards development, but making constructive progress.