RISC-V®

State of the Union

Krste Asanovic
Chief Architect, RISC-V International

RISC-V Summit, Munich
June 25, 2024

RISC-V®

@risc_v

The State of the Union is strong!

RISC-V solidly established in embedded space

On the verge of widespread adoption of RISC-V application processors

RISC-V is *the* standard base core in new AI accelerators

RISC-V: Flexible or Rigid?

- RISC-V design enables a very flexible architecture with many optional extensions
  - Supports highly tuned application-specific designs where implementer is willing to develop custom software

- RISC-V design also enables a very tightly controlled architecture with many mandates and few options
  - Supports a large software ecosystem with extensive third-party binary software
  - Profiles layer a rigid architecture definition on top of the catalog of extensions

- If you’re worrying about RISC-V fragmentation, you’re confusing the former with the latter

Binary Software Ecosystem Dynamics

- Software ecosystems want to run on largest number of devices, so target “lowest common denominator” of fielded systems
- Hardware vendors don’t want to add features that won’t be used
- To remain competitive, architectures must support a growing set of features over time

- RISC-V profiles provide a resolution to this dilemma
  - An agreed roadmap of ISA features for hardware implementers to provide in each generation, so software can rely on widespread availability and exploit these features

RVA: Application-Processor Profiles

- RVA supports rich application-processor software ecosystems
  - Examples including Linux distros, Android, …
  - RVA goal that new versions can run all software that ran on old versions
  - While many RVA sockets require the highest-performance cores (datacenter, laptop, mobile), others require low-cost cores with full feature compatibility (consumer devices)
  - ***Note: new external-facing names for RVA profile releases still under discussion***

- Initial ratified RVA profile specifications
  - RVA20 (retrospectively captures de-facto c.2016 standard ”RV64GC”)
  - RVA22 (significant clean up and multiple advances over RVA20)

- Upcoming RVA23 profile, key features:
  - Mandatory vectors
  - Mandatory hypervisor
  - High-performance vector crypto as localized option

Profiles: Mandates + Options

- RVA profiles contain mandatory extensions plus four kinds of options:
  - **Localized options:** required for different jurisdictions (e.g., crypto)
  - **Development options:** new extension in early part of lifecycle
  - **Expansion options:** large implementation overhead and not always needed, but can be handled via runtime discovery (e.g., matrix engine)
  - **Transitory options:** not clear if will remain in profile or be dropped (e.g., scalar crypto)

Major versus Minor RVA Releases

Discussion is ongoing, but tending towards:

- **Major** releases signify substantial new mandatory functionality
  - RVA20 initial “major” release
  - RVA23 next major release (vector+hypervisor mandatory)
- Future **minor** releases only introduce options, no mandates
  - Ensures new standard software continues to run on hardware from previous major release
  - While enabling hardware and software support of new features (e.g., development options that will become mandates)

Profiles to Platforms

Profiles only cover ISA features, full hardware platforms need much more.

Great progress on standard server platform specifications in Server SOC, BRS groups.

Full server platform spec expected soon.

Will provide standard for RISC-V Certification program.

Specification Improvements

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

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 and github repo to give corrections, feedback, and input on future document tree structure.
- Plan is to officially version and publish the entire spec at some frequency as new features are ratified
  - All new specs to be developed as branches on this repo, so full impact on current spec can be evaluated more easily

RISC-V and AI

- RISC-V originally developed in 2010 to support specialized computing engines, *before* current wave of AI interest
- RISC-V standard vectors efficient at mixed-precision operations, as needed in AI
- RISC-V already adopted in many AI accelerators, from Nvidia to Facebook
- RISC-V flexibility supports specialized AI extensions

- What about standard AI-specific extensions beyond vectors?

RISC-V Matrix Extensions

Two types of standard matrix extensions being developed:

- Integrated Matrix Extension (IME)
  - Matrix operations using vector registers as input and output state
- Attached Matrix Extension (AME)
  - Additional matrix state used to hold accumulators to support higher throughput

Standardize ISA or API for Matrix Operations?

- There are relatively few high-arithmetic-intensity 2D matrix kernels that benefit from matrix acceleration
- There will be a great variety in hardware support (none, vectors, IME, AME, custom, …)
- How performance-portable are single binary implementations of matrix operations?
- Different optimized libraries behind same API needed to support all targets anyway?

New Security Extensions in Progress

- Supervisor Domains (Smmtt)
  - Flexible support for confidential computing and other applications
- Lightweight Memory Tagging
  - Improve memory safety, building upon pointer masking support
- CHERI
  - Bringing capabilities to RISC-V
- Additional Crypto
  - PQC, and other vector crypto enhancements

Summary

- RISC-V is well-established in embedded sockets
- RISC-V maturing into reliable partner for application-processor software ecosystems
- RISC-V specifications improving into single coherent navigable document tree
- Significant progress towards standard server platforms with RAS, QoS, security features.
- RISC-V already considered the natural starting point for new AI accelerators – how best to intersect with evolving standard AI software stacks?