09 April 2025

CHERI

CHERI-RISC-V Standardisation

Memory safety for all, from ear-buds to servers

Tariq Kurd
Chief Architect, Codasip

RISC-V is about to have 7 base architectures

- New **unratified** CHERI base architectures are:
  - RV64CH – 32 registers, 64-bit, 128-bit capabilities
  - RV32CH – 32 registers, 32-bit, 64-bit capabilities
- Current **ratified** base architectures are:
  - RV64I – 32 registers, 64-bit
  - RV64E – 16 registers, 64-bit
  - RV32I – 32 registers, 32-bit
  - RV32E – 16 registers, 32-bit
- The other **unratified** base architecture is:
  - RV128I – 32 registers, 128-bit

Documentation status

- The current CHERI spec 0.9.5 on GitHub is 278 pages
  - ~ 170 are instruction pages
- All the features are in one document
  - This have been split into different pieces
    - Unprivileged architecture – RV64CH, RV32CH
    - Privileged architecture – Smcheri, Sscheri
    - Debug architecture – Sdcheri
  - All are currently under ARC review

Being a base architecture is useful

- It means that we can choose what we import from RV64I/RV32I.
- In particular we can ditch the fairly useless double precision floating point load/store encodings and have useful ones instead.

RV32I/ RV64I	Description	RV32CH/ RV64CH	Description
C.FSD	Store double via rs1	C.SC	Store capability via cs1
C.FLD	Load double via rs1	C.LC	Load capability via cs1
C.FSDSP	Store double via stack pointer	C.SCSP	Store capability via stack pointer
C.FLDSP	Load double via stack pointer	C.LCSP	Load capability via stack pointer

Unprivileged Architecture RV32CH/RV64CH

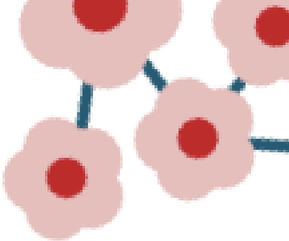
- This is the definition of the pure-capability machine
- 32 CLEN-wide registers (extended from XLEN-wide)
- All CHERI instructions are listed in the base architecture
  - E.g. load/store capability LC/SC
  - Capability manipulation
- Application Code programmer’s view
- No hybrid features
  - DDC, mode switch

Privileged Archiecture Smcheri/Sscheri

- This includes all pure capability machine CSRs visible from M or S-mode
  - E.g. extra bits in Xenvcfg, mseccfg
- Virtual memory changes
  - PTE.CW – to restrict capability writes
  - PTE.UCRG – should this be a separate extension?
- CHERI exception types and handling
  - Usage of Xtval2 for example

What else?

- The encoding format is now in an appendix
  - They’re different for CHERIoT so need to be optional
  - We need to describe the instructions without the format
- Hybrid mode is an option on the base RISC-V architecture
  - Predicated on implementing RH32CH/RV64CH
- The debug specification needs updating
  - CHERI does affect debug mode
  - Sdcheri will be pushed to the debug specification



Thanks to the contributors

- Writing the RISC-V spec

Codasip

Google

UNIVERSITY OF CAMBRIDGE

- Reviewing the RISC-V spec

RISC-V®

SCI Semiconductor

lowRISC

CHERI

THANK YOU

Contact Tariq.Kurd@codasip.com

Web www.codasip.com

Codasip

This work © 2025 by CHERI Alliance is licensed under CC BY-SA 4.0 (Creative Commons Attribution-ShareAlike 4.0 International) – https://creativecommons.org/licenses/by-sa/4.0/

09 April 2025