CHERITech'26
CONFERENCE

CHERI

12–13 Nov 2026
Munich, DE

CVA6-CHERI: (Nearly) Ratified Specification Implemented and Vector Support Planned

Moving to RVY 0.9.10

Jonathan Woodruff
Peter Rugg and Alexandre Joannou
Capabilities Limited



Located at semi | SEMICON EUROPA

Why CVA6-CHERI?

1 Research cores

Years of CHERI processors in Bluespec SystemVerilog: lessons on what works and what doesn't

2 Open reference

CVA6-CHERI: an ASIC-ready, open-source demonstrator of the RVY specification

3 Commercial CHERI

Companies build RVY cores with a working reference to compare against

Why CVA6?

SystemVerilog

Industrial-style RTL at the OpenHW Group

Adopted

Used by several companies

Active

Many contributors, ~29 commits/month

Hypervisor

Early open-source demonstrator of the H extension

Pulling the whole ecosystem forward to RVY 0.9.10

0.9.3 / 0.9.4 1.7 years ago

Ecosystem aligned: CVA6-CHERI, Codasip, QEMU, LLVM

0.9.8.2 Summer 2026

New encodings: every RVY instruction under one major opcode

0.9.9 Sept 2026

New ecosystem alignment: Sail, QEMU, LLVM, OpenSBI, CheriBSD and CVA6-CHERI all updated

0.9.10 Now

Quick ecosystem realignment Boots CheriBSD on CVA6-CHERI FPGA

Everything moves together

Hardware

CVA6-CHERI · Debug unit

Model, simulator, verification

Sail · QEMU · TestRIG

Toolchain & Debugging

LLVM · OpenOCD

Software

OpenSBI · U-Boot · CheriBSD · CHERI-seL4 · CHERI Linux

Monumental Effort!
Thank you,
CHERI Research Centre

Change 1: capability jump faults move to the target

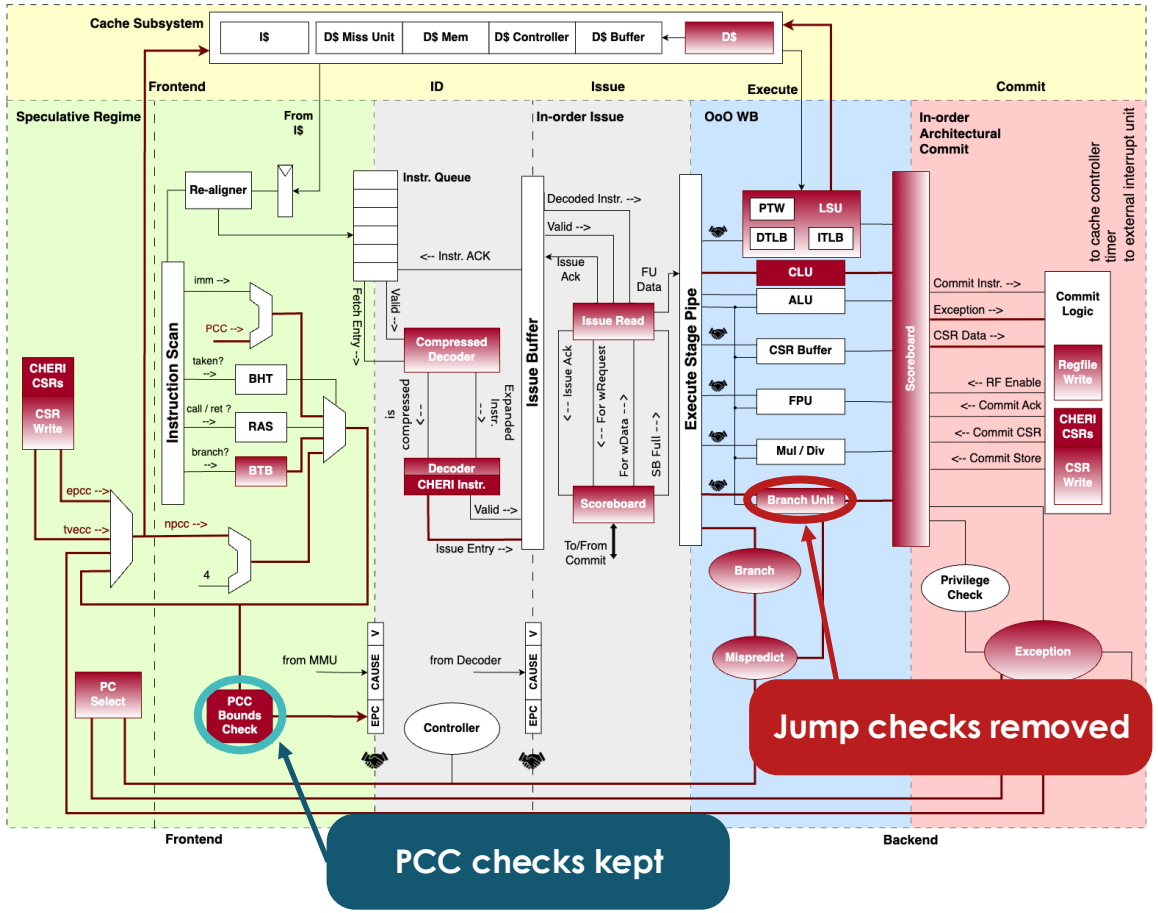
Before: trap at the jump

Jumping to an untagged or illegally sealed capability raised the exception on the jump instruction

RVY 0.9.10: trap at the target

The jump completes; the fault is taken on fetch from the bad PCC, like other RISC-V bad-target faults
(e.g. unmapped addresses)

- **PCC checks were needed anyway** (e.g. capability installed in xEPC, then xRET)
- **Removed the duplicate checks** from the branch unit
- **Less logic and a shorter critical path**



Change 2: richer page-table capability controls

Table 27. Summary of Load CW and CRG behavior in the PTEs

PTE.CW	PTE.U	PTE.CRG	Load/AMO
0	X	X	Clear loaded tag
1	1	≠ sstatus.UCRG	Page fault, or page fault if tag is set[1]
1	1	= sstatus.UCRG	Normal operation
1	0	X	Normal operation[2]

RVY 0.9.3

RVY 0.9.10

Table 53. Breakdown of the 4-bit pte.rvy field.

Bit	Name	Comment
Capability load/AMO fields		
pte.rvy[0]	pte.yr	Capability readable
pte.rvy[1]	pte.yrg	Capability read generation
Capability store/AMO fields		
pte.rvy[2]	pte.yw	Capability writable
pte.rvy[3]	pte.yd	Capability dirty

sstatus.YRGE

1

0

Table 51. Definition of the pte.rvy field

Bit	Name	Comment
pte.rvy[3]	pte.y	Enable capability (Y) access
pte.rvy[2]	reserved	reserved
pte.rvy[1]	reserved	reserved
pte.rvy[0]	reserved	reserved

Enables kernel-controlled generational revocation

sstatus enables

piped through the core

→

PTW / TLB / LSU

check PTE state

→

Capability access

allow · strip tag · fault

What it took

- **Plumb new CSR enables** through translation and memory access
- **Align exception priorities** with Sail
- **Resolve spec ambiguities** with Sail
- **Update CheriBSD** to use new states for revocation (tested on QEMU first, worked on CVA6-CHERI)

Bring-up example issue early in OpenSBI

```
UART: (blank screen)
```

```
Attached with OpenOCD+GDB:
mcause = illegal instruction
PC = X
```

```
X: beq a0, a1, ...
```

~20th instruction of OpenSBI

Debug unit + OpenOCD showed exactly where it died

- **Also disagreements** around PTE fault priorities
- **Converged** Sail, QEMU, LLVM and CVA6-CHERI

Sail + CVA6-CHERI

Draft spec: branches with rs1 ≤ rs2 reserved (redundant operand order)

Agree via TestRIG

≠

LLVM + QEMU + CheriBSD

Compiler authors never implemented the reservation

Boots happily in QEMU

Resolution: align with LLVM and QEMU —the Architecture Review Committee disliked the reservation too!

Current status: RVY 0.9.10 boots on FPGA

0.9.10

RVY revision implemented, ~1.7 years newer than 0.9.3

FPGA

Booting CheriBSD, seL4, and prototype Linux on FPGA
Improved Tag Controller
Passes timing on Genesys2

Mocha

Integrated into lowRISC Mocha: core update with floating point and caches enabled

Running on the new ISA

OpenSBI | U-Boot | CheriBSD + tests | CHERI-seL4 | CHERI Linux* | LLVM | QEMU | Sail

**AI-assisted port from the previous CHERI ISA*

Community patches: eight from John Goodacre

Built Mocha for FPGA and ASIC flows → 8 patches with detailed justification → 7 commits

1 Fetch flow control

Root-caused a lock-up fetching from non-idempotent memory: an instruction dropped when flow control was ignored

2 AES in superscalar

The upstream AES option only worked single-issue; fixed port assignment and blocking

3 Faster bounds decode

Precompute both addr_hi cases for GetTop/GetBase; ~5% area if applied everywhere, so only on critical paths

4 Jump-to-sentry

Back-ported 0.9.10's trap-at-target from our old 0.9.3 core for timing … and exposed something we had missed

One bug, found three independent ways

1 Code review

John Goodacre's back-port had sentry conditions our RTL lacked

2 Generated tests

Alex Richardson generated tests for the RVY ratification effort that discovered a failure in this case for the executable RVY Sail specification

3 Formal proof

AIVHAI teams, including Ben Laurie, proving CHERI properties on Sail found it in a violation of monotonicty—plus a corner our fix missed

The bug

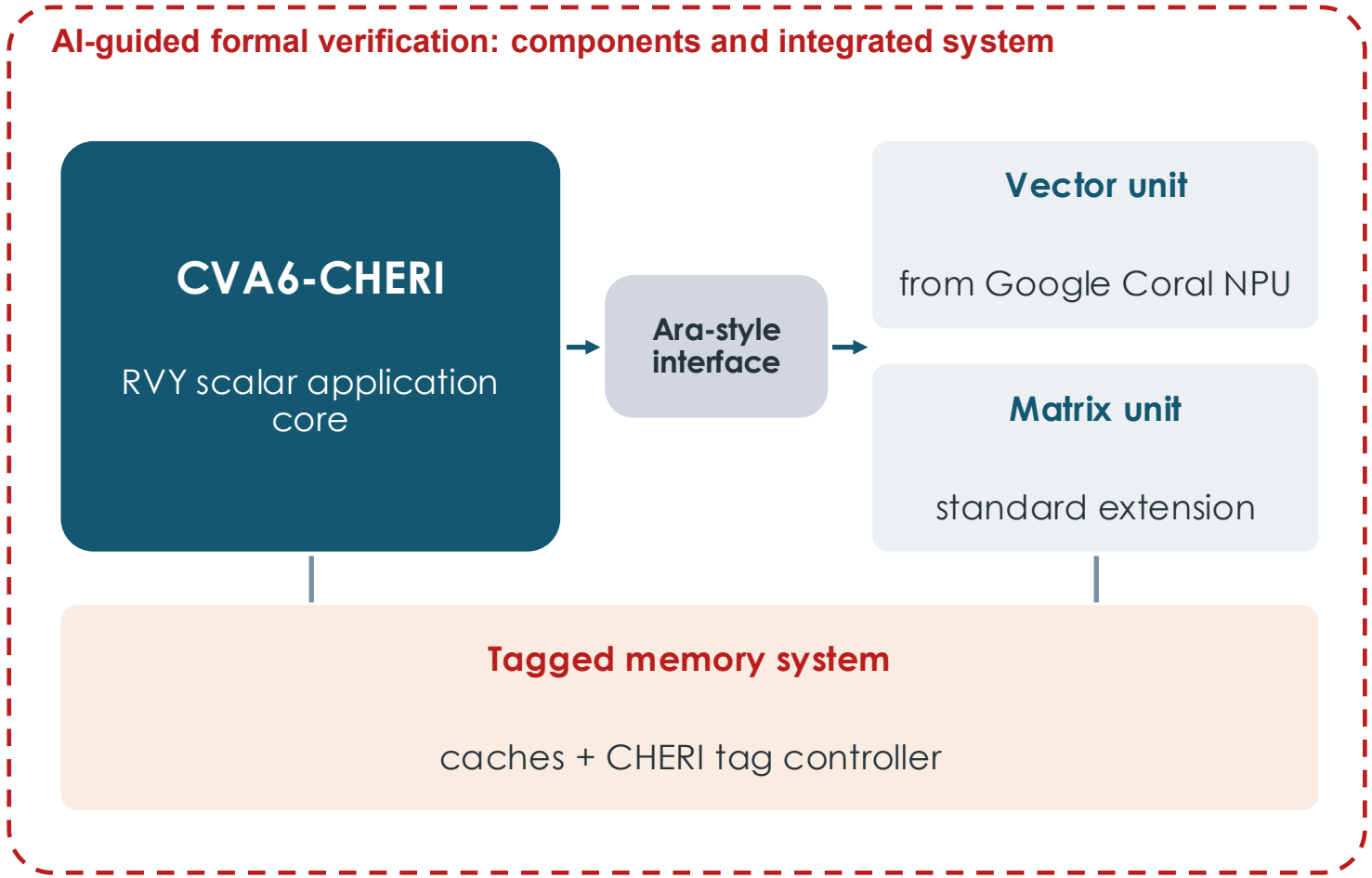
Sentry jump conditions missing in both Sail and RTL, e.g. jumping to a sentry at a non-zero offset must clear the target's tag

Risk: in a corner case, a usable sealed pointer without unsealing — a compartment bypass

Fixed in Sail and CVA6-CHERI

Also: AIVHAI red-team reports (e.g. YPERMC on sealed capabilities) drove further Sail and RTL fixes

Next: vector and matrix for a CHERI AI accelerator



AIVHAI (ARIA-funded)

- Standard RISC-V **vector and matrix** extensions
- Hybrid of the **PULP Ara** interface and **Coral NPU** vector unit
- Goal: a **memory-safe AI accelerator** for embedded systems
- Testbed for **AI-automated formal verification**

Summary



Spec Improvements

- RVY 0.9.10 in RTL, Sail, QEMU, LLVM and software
- New Ecosystem Boots on FPGA



Community Verification

- Simpler jump checks; richer PTE enforcement
- Fixes from community, formal proofs and red team



New Directions

- Vector and matrix units via AIVHAI
- AI-guided formal verification of the whole system

CHERITech'26 CONFERENCE

12–13 Nov 2026
Munich, DE



THANK YOU

Contact Jonathan.Woodruff@capabilitieslimited.co.uk

Web www.capabilitieslimited.co.uk/projects/cheri-cva6

Code github.com/Capabilities-Limited/cheri-cva6

Located at semi | SEMICON EUROPA

CHERI™ and the CHERI logo are protected trademarks. Use of these marks is subject to the CHERI Alliance Trademark Policy – https://cheri-alliance.org/legal-and-policies/trademark/

This work © 2026 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/