CHERI RISC-V: A Case Study on the CVA6

Bruno Sá, Donato Ferraro, José Martins, Everton Matos, Andrea Bastoni and Sandro Pinto

RISC-V®

RISC-V Summit Europe 24 @ Munich

FCT Fundação para a Ciência e a Tecnologia

GOVERNO DE PORTUGAL | MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Universidade do Minho

CENTROALGORITMI

MINERVASYS

ZERO DAY LABS

TII Technology Innovation Institute

Jun 27th, 2024

Agenda

Introduction

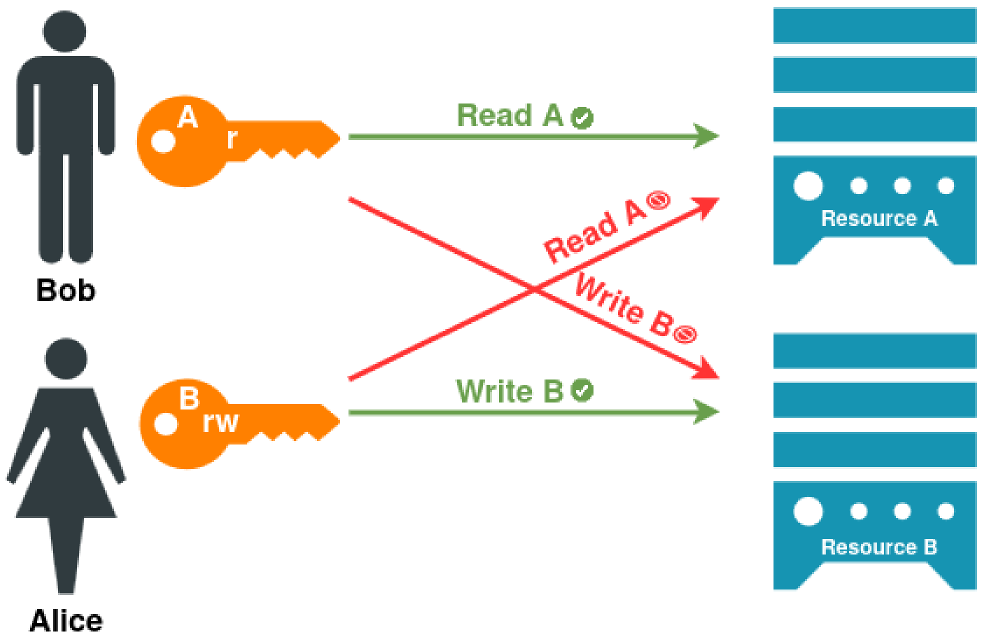
Capability-based Security and CHERI RISC-V Extension

Capability-Based Security

Capabilities are unforgeable key of authority

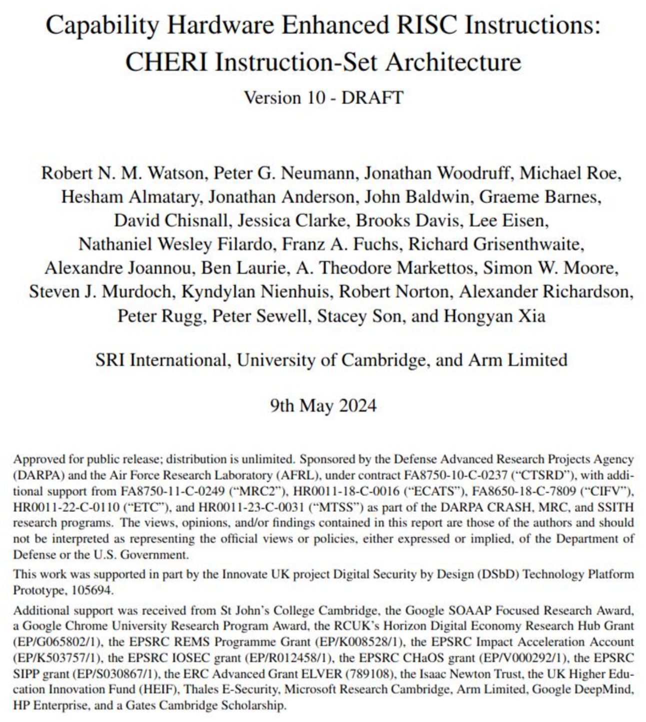
- Based on the principle of least privilege
- Secure and controlled sharing of resources
- Deterministic protection
- Prevents the confused deputy problem
- To access a resource a user must present a valid capability to it
- Been around for decades:
  - M-Machine, Intel iAPX 432, etc
  - Hydra, Sel4 and Google Fuchsia

Capability-based Access



CHERI RISC-V Extension In a Nutshell

- **Boosts the ISA with capability-based primitives**
- **Based on two security principles:**
  - Principle of least privilege
  - Principle of intentional use
- **Fine-grained memory protection**
- **Scalable software compartmentalization**
- **Deterministic intra-address protection**
- **Architectural Capabilities (ptrs as caps)**
- **Defines a set of instructions to manipulate capabilities**
- **Adds a set of capability-aware per-mode SCRs:**
  - *pcc, ddc , <m/s>tcc <m/s>tdc*



Capability Hardware Enhanced RISC Instructions:
CHERI Instruction-Set Architecture
Version 10 - DRAFT

Robert N. M. Watson, Peter G. Neumann, Jonathan Woodruff, Michael Roe, Hesham Almatary, Jonathan Anderson, John Baldwin, Graeme Barnes, David Chisnall, Jessica Clarke, Brooks Davis, Lee Eisen, Nathaniel Wesley Filardo, Franz A. Fuchs, Richard Grisenthwaite, Alexandre Joannou, Ben Laurie, A. Theodore Markettos, Simon W. Moore, Steven J. Murdoch, Kyndylan Nienhuis, Robert Norton, Alexander Richardson, Peter Rugg, Peter Sewell, Stacey Son, and Hongyan Xia

SRI International, University of Cambridge, and Arm Limited

9th May 2024

Approved for public release; distribution is unlimited. Sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237 ("CTSRD"), with additional support from FA8750-11-C-0249 ("MRC2"), HR0011-18-C-0016 ("ECATS"), FA8650-18-C-7809 ("CIFV"), HR0011-22-C-0110 ("ETC"), and HR0011-23-C-0031 ("MTSS") as part of the DARPA CRASH, MRC, and SSITH research programs. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Department of Defense or the U.S. Government.

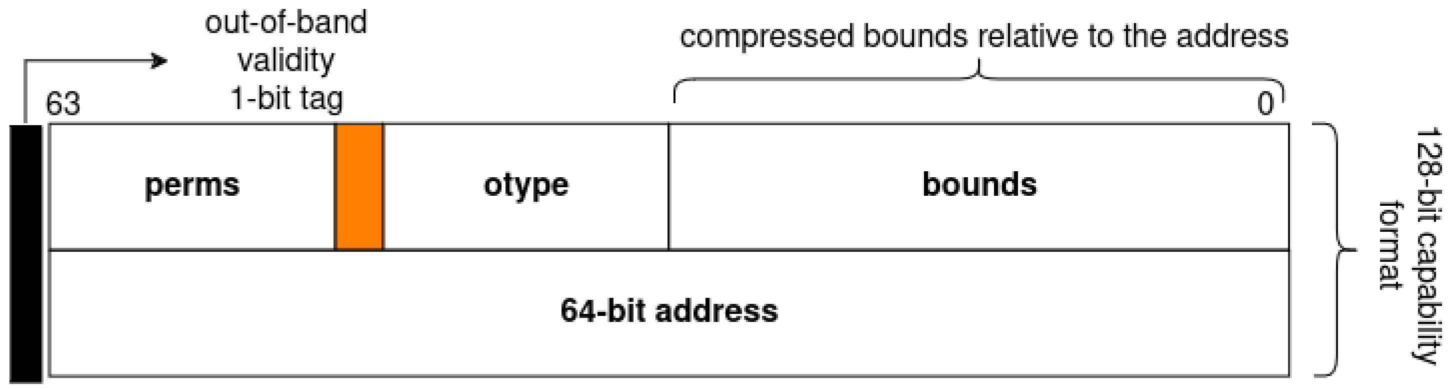
This work was supported in part by the Innovate UK project Digital Security by Design (DSbD) Technology Platform Prototype, 105694.

Additional support was received from St John's College Cambridge, the Google SOAAP Focused Research Award, a Google Chrome University Research Program Award, the RCUK's Horizon Digital Economy Research Hub Grant (EP/G065802/1), the EPSRC REMS Programme Grant (EP/K008528/1), the EPSRC Impact Acceleration Account (EP/K503757/1), the EPSRC IOSEC grant (EP/R012458/1), the EPSRC CHaOS grant (EP/V000292/1), the EPSRC SIPP grant (EP/S030867/1), the ERC Advanced Grant ELVER (789108), the Isaac Newton Trust, the UK Higher Education Innovation Fund (HEIF), Thales E-Security, Microsoft Research Cambridge, Arm Limited, Google DeepMind, HP Enterprise, and a Gates Cambridge Scholarship.

https://github.com/CTSRD-CHERI/cheri-specification/releases

CHERI Architectural Capabilities

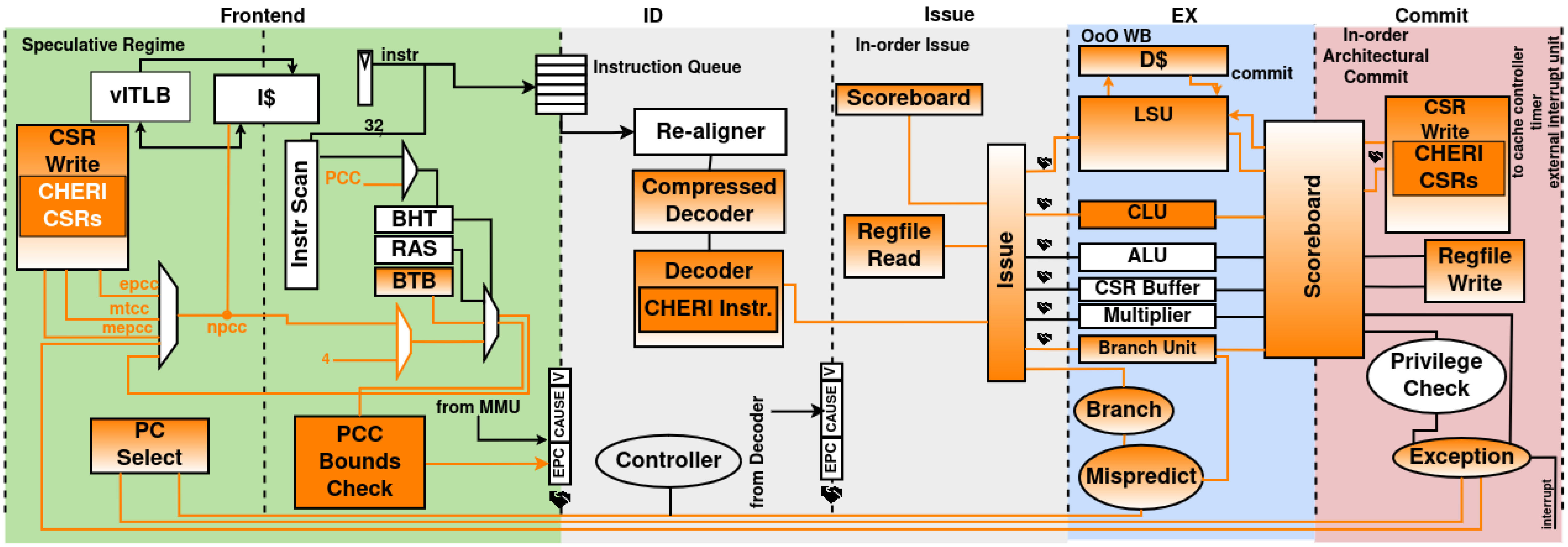
- Mitigates most vulnerability exploitation related to spatial memory safety (e.g., out-of-bounds access)
- Every memory access enforces bounds, permissions checks
- Capabilities guarantee valid provenance and pointer integrity via 1-bit tag
- ISA instructions enforce monotonicity and guarded manipulation
- Object-capability model support


out-of-band validity 1-bit tag
compressed bounds relative to the address
63
0
perms
otype
bounds
64-bit address
128-bit capability format

CVA6H-CHERI Support

Overview, Features, Status, Implementation and Hardware Overhead

CVA6H-CHERI: Overview



CVA6H-CHERI: Features and Status

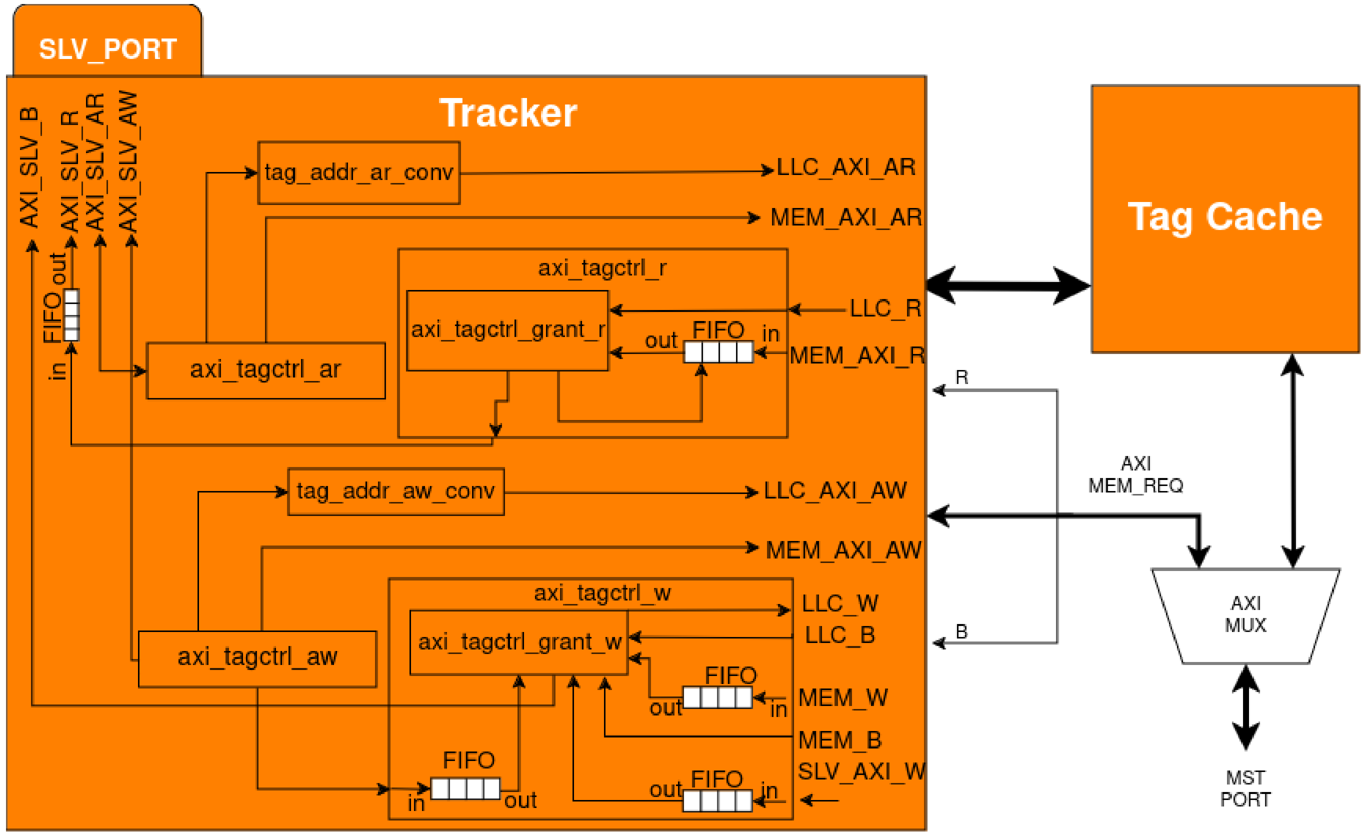
- CHERI ISA v9 Cambridge
- Moving to RISC-V CHERI standard
- RV64 and RV32 (MMU or MMU-less)
- Merged Capability Register File
- Integer and Capability mode supported
- DDC (Default Data Capability) support
- Tagged-memory support
- Compressed Capability Format
- CHERI CSRs
- Capability-Manipulation Instructions
- CHERI & Hypervisor extension
    - CSRs (vstcc, vsepcc, vstdc)
    - HS VM LD/ST Cap Instructions (hlv/hls)
- Optional extension

CHERI CSRs	Description	Extends	Status
PCC	Program Counter	PC	✓
DDC	Default Data Capability		✓
<m/s>tcc	Trap code capability	*<m/s>tvec*	✓
<m/s>tdc	Trap Data Capability		✓
<m/s>epcc	Trap exception PC capability	*<m/s>epc*	✓
<m/s>ScratchC	Scratch capability		✓

CHERI Inst.	Status
Cap.-Inspection/-Modification	✓
Pointer-Comparison/Arithmetic	✓
Control-Flow	✓
Fast Register-Clearing Instr.	x
Adjusting to Compressed Capability Precision Instructions	✓
Tag-Memory Access Instructions	x
Memory Load/Stores Capability via Capability	✓

CVA6H-CHERI: AXI TagController

- 1-bit tag per capability
- No tag compression support (yet)
- Stores tags in Table located at reserved space on DRAM
- Parameterizable size
- AXI4 support
- TagCache based on Pulp axi_llc*



* https://github.com/pulp-platform/axi_llc

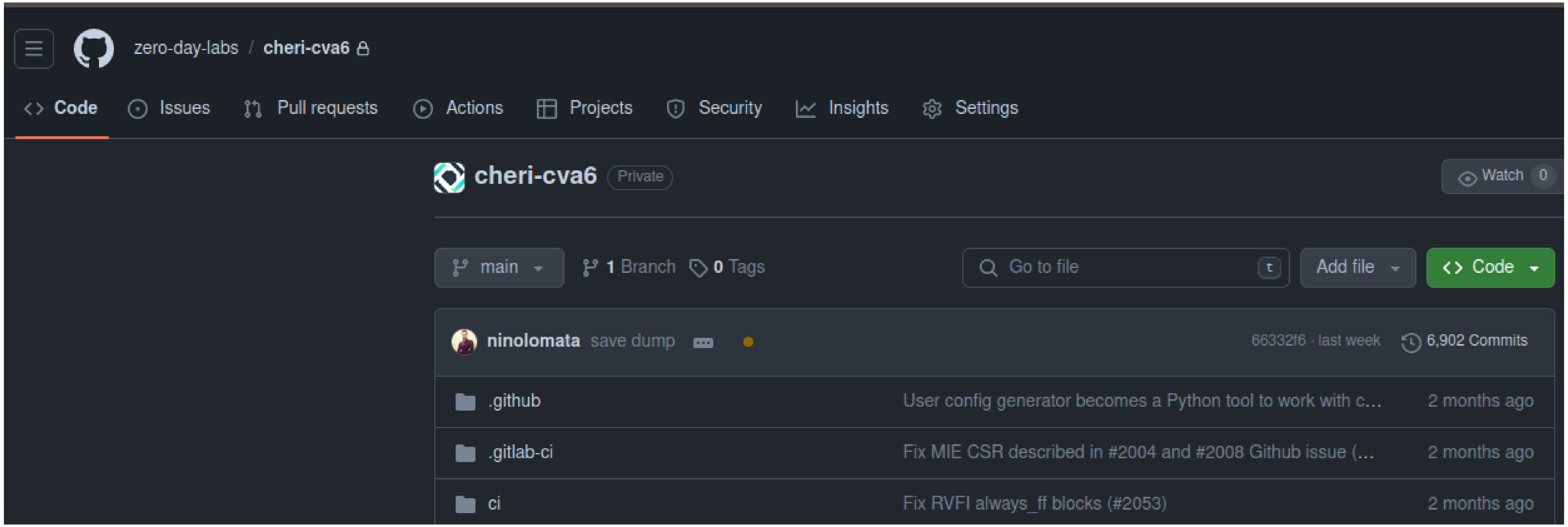
CVA6H-CHERI: Validation (WiP)

- **TestBenchs Simulation (Done)**
  - Unit Tests Cheri Logic Unit, Branch Unit, and Load/Store Unit
- **FPGA Emulation (Done)**
  - Experimental SoC with Tagged Memory support
- **Baremetal Tests (WiP)**
  - Handwritten C Tests built atop the riscv-hyp-bare *
- **TestRig (WiP) ****
  - Add CVA6H-Cheri implementation to TestRig Framework **(Done)**
  - Implementing RVFI-DII **(WiP)**
- **Bao Hypervisor (WiP)**
  - Started porting Bao hypervisor to CHERI using QEMU
- **CheriBSD (Not Started)**

** https://github.com/josecm/riscv-hyp-tests*
*** https://github.com/ninolomata/TestRIG*

CVA6H-CHERI: Implementation

- Fork of CVA6 with hypervisor extension support + cheri*
- Optional extension via config parameter *CVA6RVZcheri*
- *FPGA emulation with experimental CVA6H-CHERI-based SoC + Tagged Memory Controller***



ninolomata save dump · 66332f6 · last week · 6,902 Commits

.github	User config generator becomes a Python tool to work with c…	2 months ago
.gitlab-ci	Fix MIE CSR described in #2004 and #2008 Github issue (…	2 months ago
ci	Fix RVFI always_ff blocks (#2053)	2 months ago

* https://github.com/zero-day-labs/cheri-cva6.git
** https://github.com/zero-day-labs/axi-cheri-tagctrl.git

CVA6H-CHERI: Hardware Overhead

- Single core CVA6H-CHERI with TagController support targeting Genesys2 FPGA (no otmizations yet)

Module	FPGA Resources (+ % overhead compared to vanilla CVA6)	
	LUTs	FFs
CVA6 SoC	101587 (+36%)	56107 (+33%)
CVA6 Core	68991 (+38%)	26284 (+31%)
wt_cache_subsystem	9963 (+23%)	3034 (+36%)
load/store unit	13818 (+12%)	9010 (+13%)
execute stage	26752 (+16%)	11756 (+12%)
cheri logic unit	1953	0
issue_stage	22366 (+72%)	4656 (+38%)
AXI Tag Controller	7948	7762

Next Steps and Roadmap

Next Steps and Roadmap

- Complete validation (Baremetal C Tests, TestRig, Bao Hypervisor and CHERIBSD)
- Port Bao Hypervisor to CHERI
- Explore CHERI software models in the context of virtualization (hybrid and pure capability)
- Bump implementation to comply with the RISC-V CHERI standard extension RV64 and RV32
- Explore trade-offs performance/hardware overhead

Virtualization reference stack with CHERI support for MCSs systems

THANK YOU!

bruno.vilaca.sa@gmail.com
https://github.com/ninolomata
https://twitter.com/ninolomata
https://www.linkedin.com/in/brunosa560/

Q&A

GET IN TOUCH WITH US!

- **Address:** Universidade do Minho, Campus Azurém, 4800-058 Guimarães, PORTUGAL
- **Phone:** +351 253 510180
- **Email:** bruno.vilaca.sa@gmail.com