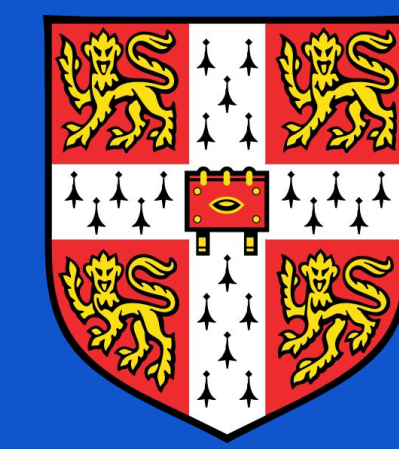


Toward industrial grade CHERI enhanced RISC-V cores

Alexandre Joannou, Jonathan Woodruff, Peter Rugg, Matthew Naylor, Franz A. Fuchs and Simon Moore
Department of Computer Science and Technology, University of Cambridge
alexandre.joannou@cl.cam.ac.uk
cheri-cpu.org



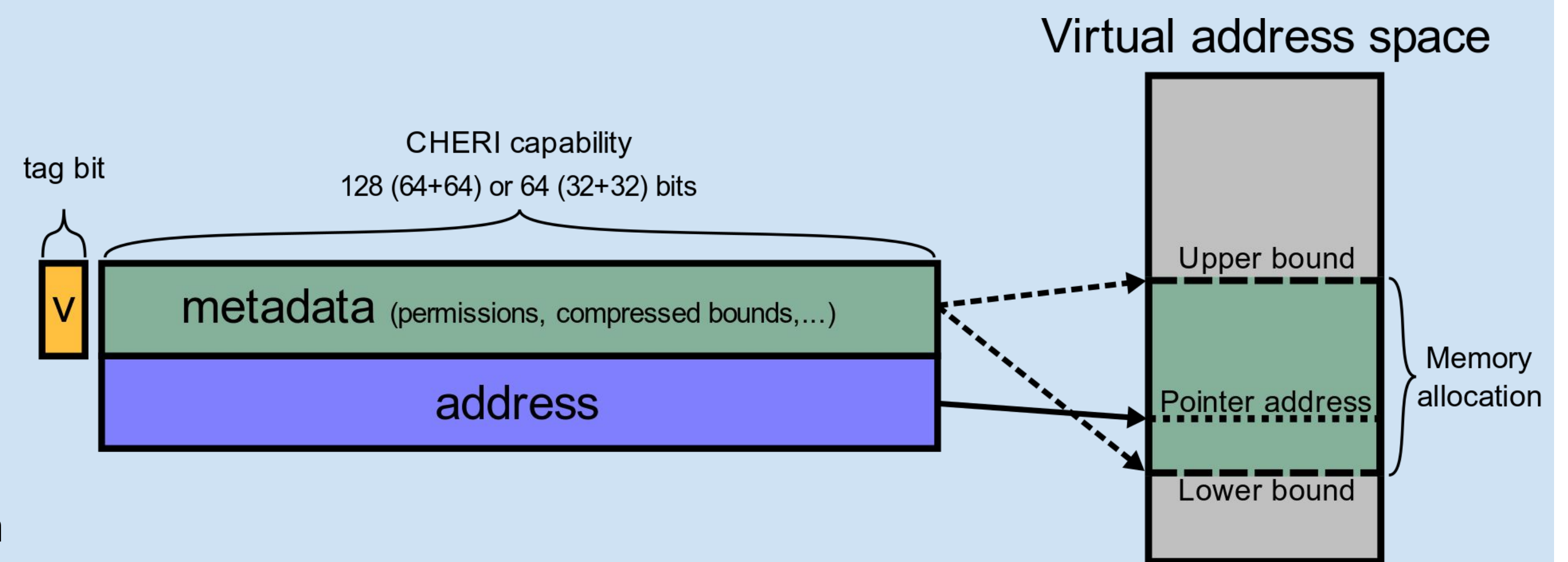
CAPABILITIES
LIMITED



UNIVERSITY OF
CAMBRIDGE
Computer Science & Technology

What is CHERI?

- A security-focused ISA extension (new hardware type: the CHERI **capability**)
- Capabilities are
 - **unforgeable** tokens of authority, stored in **tagged** memory and registers
 - used and stored like conventional pointers
 - manipulatable straight from **user-space** (no expensive privilege escalation mechanism needed)
 - only **monotonic decrease** in privilege is possible (no privilege escalation possible)
- Every memory access is checked against a capability's covered memory region and access rights. Capability access checks are **non-bypassable**.



bluespec



The time-tested CHERI-cap-lib

- Reference CHERI functionality implementation for hardware design
 - Used in many cores
 - Backed all our CHERI publications
- BUT*
- Written in **Bluespec SystemVerilog**, not in SystemVerilog

SystemVerilog

Recent effort in industrial context

- **SystemVerilog** CHERI functions in **CVA6**
- CVA6 is generally considered an industrial-grade reference RISC-V core

Exploratory work extending CVA6 with CHERI by **Zero-Day Labs**

Formally **verify equivalence** between SV and BSV functionality

bluespec



SystemVerilog

A trusted CHERI-cap-lib SV implementation

- A SystemVerilog version derived from **Zero-Day Labs**' work, updated according to verification results, implementing the University of Cambridge ISAv9 CHERI ISA
- A SystemVerilog version implementing the **Zcheri** proposed standard, developed via equivalence checking only against the original BSV version, to be also used in CVA6

- Use generated Verilog wrappers around Bluespec functionalities
- Use SystemVerilog implementation functions
- Provide a library of SystemVerilog modules comparing individual functionalities' results
- Use **SymbiYosys** for formal verification of the equivalence
- Enable verification of changes to CHERI-cap-lib in continuous integration

Additional work

Some further work being conducted to advance CHERI adoption in the industrial context:

- Running CVA6 through a **TestRIG** flow (TestRIG generates instruction sequences and compares execution trace with the **Sail** golden model)
- Implementing an efficient, hierarchical **tag cache**, making use of the **hpdcache**, extended to support single-bit words
- Update the CVA6 pipeline where needed to make use of the Zcheri SystemVerilog version of CHERI-cap-lib
- Check **formal properties** of the encoding format of CHERI capabilities (currently done on Bluespec) in addition to equivalence checking between SystemVerilog and Bluespec.