CHERI Standardisation

Prof. Simon W. Moore

Approved for public release; distribution is unlimited. This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contracts FA8750-10-C-0237 ("CTSRD"), with additional support from FA8750-11-C-0249 ("MRC2"), HR0011-18-C-0016 ("ECATS"), FA8650-18-C-7809 ("CIFV"), HR001122C0110 ("ETC"), and HR001122S0003 ("MTSS"). The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Approved for public release; distribution is unlimited.

The original CHERI R&D was supported 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”), HR001122C0110 (“ETC”), HR001123C0031 (“MTSS”), and FA8750-24-C-B047 (“DEC”) as part of the DARPA I2O CRASH, I2O MRC, MTO SSITH, and I2O CPM 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 of the Department of Defense or the U.S. Government.

Arm’s Morello and significant recent developments in the CHERI software and hardware stacks were supported in part by the Innovate UK project 105694 (“Digital Security by Design (DSbD) Technology Platform Prototype” and Innovate UK project 10027440 (“Developing and Evaluating an Open-Source Desktop for Arm Morello”).

We further acknowledge EPSRC REMS (EP/K008528/1), EPSRC CHaOS (EP/V000292/1), ERC ELVER (789108), the Isaac Newton Trust, the UK Higher Education Innovation Fund (HEIF), Thales E-Security, Microsoft Research Cambridge, Arm Limited, Google, Google DeepMind, HP Enterprise, and the Gates Cambridge Trust.

Introduction

- CHERI is a portable security model
- Requires support at the Instruction Set Architecture (ISA)
  - Interface between hardware and software
  - But portable between ISA models: Arm, x86 (Intel/AMD), RISC-V, etc.
- Current CHERI support in ISAs
  - Proprietary ISA: ARM Morello experimental platform
  - Open-source ISA: CHERI for RISC-V

ARM Morello

- Innovate UK Digital Security by Design program has allowed many companies to successfully evaluate CHERI technology
  - https://www.dsbd.tech/
- Many UK companies would like to purchase CHERI hardware, but none have a big enough market

Configured THINGS
cambridge consultants Part of Capgemini Invent
arm
THALES
AdaCore
ULTRA Intelligence & Communications
CHEVIN TECHNOLOGY
science scope
SCI Semi
lowRISC
Google
Microsoft
OXON TECH MANAGED IT SERVICES & SUPPORT
kaze
SGN
pytilia
DATACORE
MBDA
hp
dynamicDevices LTD
Metrarc
deltaflare
AMPLIPHAE
rtegrity
goldilock
Cyber Defence Service
REALVNC
INVENTIA UK
IOETEC
cyntegra fast | local | complete | recovery
MISSION CRITICAL APPLICATIONS
Sensorit
openlx SPECIAL PROJECTS
prizm
THG
ExactTrak
trèsbian
RISKOA SECURING A SAFE WATER FUTURE
vividgrd.

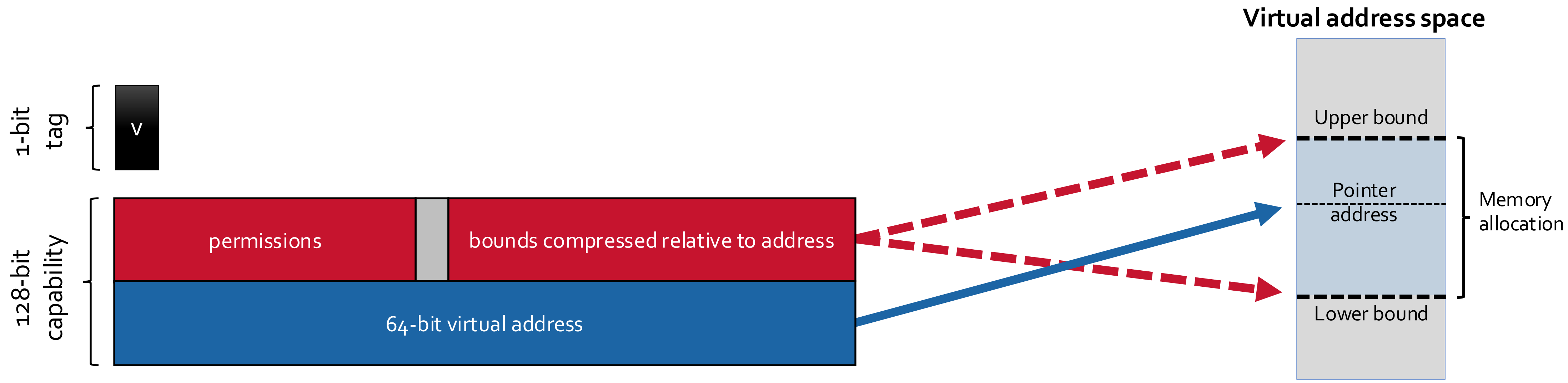
Visible commercial hardware activity

- Codasip announced CHERI cores at the RISC-V Summit North America 2023
- Microsoft has released CHERIoT - an embedded variant of CHERI
  - Sonata CHERIoT platform from NewAE and lowRISC now available through Mouser: https://media.newae.com/datasheets/NAE-SONATA-ONE_datasheet.pdf
  - Intent to manufacture CHERIoT silicon announced by SCI Semiconductor: https://scisemi.com
  - Recent RISC-V blog: "CHERIoT: A Study in CHERI": https://riscv.org/blog/2024/08/cheriot-a-study-in-cheri/
  - Microsoft released a 42-page security evaluation report on CHERI in 2023: https://www.microsoft.com/en-us/research/publication/cheriot-rethinking-security-for-low-cost-embedded-systems/
- ARM has produced the Morello (CHERI-ARM) desktop demonstrator
  - HotChips 2022: https://hc34.hotchips.org/assets/program/conference/day1/Academia/HC2022.Arm.RichardGrisenthwaite.v1_0.pdf
- Google Research is open-sourcing a CHERI-enabled ML accelerator
- There is much more activity that is still confidential

CHERI-RISC-V standardisation at RISC-V International

- CHERI Task Group has been setup
  - Alex Richardson (Google) and Simon Moore (Cambridge) co-chairing
- CHERI-RISC-V refines the CHERI architecture embodied in ARM Morello
- Substantial work has already been undertaken by industry and academia to draft a CHERI-RISC-V spec:
  - GitHub repository: https://github.com/riscv/riscv-cheri
  - HTML version of latest draft: https://riscv.github.io/riscv-cheri/
  - CHERI Alliance members including Codasip and University of Cambridge have made major contributions to the spec.

CHERI 128-bit capabilities extend integer registers



- **Capabilities** extend **integer memory addresses**
- **Metadata** (bounds, permissions, …) control how they may be used
- **Guarded manipulation** controls how capabilities may be manipulated; e.g., **provenance validity** and **monotonicity**
- **Tags** protect capability integrity/derivation in registers + memory

Summary of new instructions

I-type instruction	
caddi	Increment the cap. address by an arbitrary offset
R-type instructions	
cadd/cmv	Increment the cap. address by an arbitrary offset
scaddr	Set address of capability to arbitrary address
acperm	Bitwise AND of mask value with architectural (AP) and software-defined (SDP) bitmap permissions
scmode	Capability set CHERI execute mode
schi	Replace capability metadata with arbitrary value. Tag always cleared.
sceq	Compares two capabilities including tag, metadata and address
scss	Tests whether the bounds and permissions of a capability are a subset of another
cbld	Build a capability from another
scbnds	Set base and length of a capability. Tag cleared if encoding cannot represent the bounds exactly
scbndsr	Set based and length of a capability. Base rounded down or length rounded up to fit the encoding.
scbndsi	Capability set bounds with 5-bit immediate length

Code point	
modesw	Switch current CHERI execution mode
2-operand instructions	
sentry	Seal capability as a sentry capability
gctag	Output the tag of the input capability
gcperm	Output the architectural (AP) and software-defined (SDP) permission fields of the input capability
gchi	Output the compressed capability metadata
gcbase	Output the expanded base address of the input capability
gclen	Output the length of the input capability
cram	Output the nearest bounds alignment that a valid capability can represent

Load/Store

- In capability-mode, the integer base register becomes the capability register
  - So, no new load/store instructions to access data
  - Vector instructions also use a capability-base register to provide bounds

- One new “width” code to access capabilities (lc, sc, lr.c, sc.c, amoswap.c)
  - Currently we use the 2x XLEN width encoding (128b for RV64, 64b for RV32) to access capabilities
  - Hypervisor memory instructions also added with new width code

Other encoding space used

- PTE bits
  - 2 or 3 bits required (RV64 only)
- 10 new CSR addresses
  - 2 debug (dinfc, dddc)
  - 2 in each of M, VS (with hypervisor), S, U (trap data capability, thread id)
- Reserved bits and values
  - 1 xcause value used for CHERI faults. Corresponding single bit of medeleg used
  - 2 bits used of xenvcfg (“dirty enable” and “register enable”)
  - 1 bit used of mseccfg (“register enable”)

Conclusions

- CHERI is a portable security model that can be applied to any commercially used microprocessor (Arm, Intel/AMD, RISC-V)
- Major software evaluation by academia and industry using the ARM Morello prototype hardware platform
- Refinements to the CHERI approach appear in the CHERI-RISC-V open-source specification
    - Ready for exploitation by start-up companies wishing to innovate
    - Opportunities for large established players to use the technology