CHERI-VP: Evaluating CHERI Early for Embedded RISC-V Systems with Virtual Prototypes

Spandan Das[1], Luca Müller[2], Khushboo Qayyum[2],
Sallar Ahmadi-Pour[1], Christoph Lüth[2], Rolf Drechsler[1,2] *

[1]Institute of Computer Science, University of Bremen, Bremen, Germany
[2]Cyber-Physical Systems, DFKI, Bremen, Germany
{spandan, sallar, drechsler}@uni-bremen.de, {luca.mueller, khushboo.qayyum}@dfki.de

Abstract

The adoption of capability-based architectures such as Capability Hardware Enhanced RISC Instructions (CHERI) in constrained RISC-V systems raises open questions regarding performance overheads, verification complexity, and practical evaluation methodologies. Virtual prototyping provides an effective means to explore these questions early in the design process, before committing to Register-Transfer Level (RTL) implementations. In this paper, we present a CHERI-enabled RISC-V Virtual Prototype (VP) targeting constrained embedded systems and demonstrate its use for early architectural evaluation. We describe VP-based verification workflows for both software and hardware and report early performance insights focusing on CHERI tagged memory management. Our experiences highlight the benefits of VPs for guiding CHERI adoption decisions and identify practical challenges, including the need for lightweight benchmarks suitable for constrained environments.

Introduction

The increasing deployment of *Internet of Things (IoT)* devices [1] in safety- and security-critical domains has made memory safety a primary design concern [2, 3]. Many embedded systems are implemented in low-level languages such as C and operate under strict resource constraints, limiting the applicability of traditional software-only security mechanisms [4]. As a result, architectural support for memory safety has gained renewed attention in the embedded RISC-V ecosystem. Capability-based architectures, in particular CHERI, provide fine-grained spatial and temporal memory safety through unforgeable, permission-carrying pointers enforced in hardware [5]. While CHERI has been extensively studied in server-class and desktop systems, its adoption in constrained RISC-V platforms raises open questions regarding hardware overheads, performance impact, and verification complexity. Addressing these questions early in the design process is crucial before committing to full RTL implementations.

VPs offer an effective means to explore such design trade-offs at an early-stage. By abstracting hardware behavior while retaining architectural fidelity, VPs enable rapid experimentation with software stacks, security mechanisms, and microarchitectural features [6]. However, existing CHERI research largely focuses on hardware implementations or operating system support [7], with limited attention to VP-based early evaluation workflows targeting constrained devices. In this paper, we present CHERI-VP, an instruction-accurate RISC-V-based VP with 32-bit CHERI support. Specifically, this paper makes the following contributions: (i) a CHERI-enabled RISC-V VP framework targeting constrained devices, (ii) a VP-based methodology for early evaluation of CHERI adoption, covering memory safety behavior, verification workflows, and performance trade-offs, and (iii) initial insights from VP-driven case studies on CHERI verification and tag-memory performance. An overview for contributions (i), and (ii) is shown in Figure 1, highlighting the VP-based approach for development of safe IoT applications, while Figure 2 shows how CHERI-VP can be utilized for exploring performance improvements for secure constrained devices.

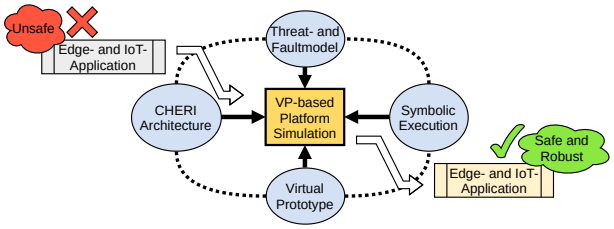

Figure 1: *CHERI-VP: Our CHERI-enabled 32-bit RISC-V VP for constrained devices.*

Verification Workflows for CHERI-enabled Systems

Ensuring correctness of CHERI-enabled systems requires verification across both software and hardware layers. While CHERI-specific verification results are still at an early-stage, our CHERI-VP enables the development and evaluation of verification workflows that can be applied to CHERI-based architectures.

* This research has been supported by the German Research Foundation (DFG) with project EMBOSOM (grant nos. DR 287/42-1, LU 707/10-1), and the German Ministry for Research, Technology and Space (BMFTR) with project ExaVerse (grant no. FKZ 16IW25003).

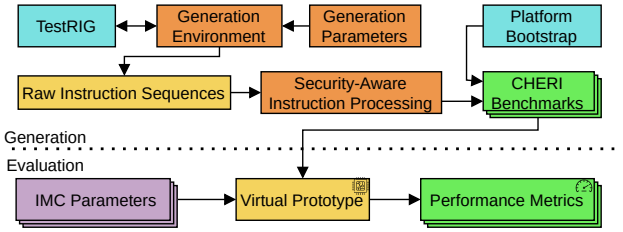

Figure 2: *VP-based exploration of IMC-accelerated architecture optimization with CHERI benchmarks.*

On the software side, we investigate the use of symbolic execution techniques to guide test generation toward memory safety violations that manifest as CHERI exceptions. Early results show the potential of CHERI-VP to uncover security vulnerabilities related to spatial and temporal memory safety violations documented in the *Common Weakness Enumeration (CWE)* [8]. Utilizing prior experience with symbolic execution for embedded software, relevant execution paths through the software to detect such violations are identified. By employing CHERI as an error detection mechanism, CHERI-VP enables early exploration targeted at the automatic detection of security vulnerabilities recorded in the CWE in real-world embedded software.

On the hardware side, CHERI-VP serves as an executable reference model for CHERI instruction semantics and capability propagation. We leverage mutation-based verification and instruction stream generation techniques to validate the functional correctness of CHERI instruction handling in the VP. These workflows form the foundation for future cross-level verification against CHERI-enabled RTL implementations.

Early Performance Insights and Optimization Exploration

Performance overhead is a key concern when adopting CHERI in resource-constrained systems. One notable source of overhead is tag memory management, which requires additional memory accesses during capability store operations to maintain non-addressable tag bits. Here, emerging techniques like *In-Memory Computing (IMC)* can help to reduce data movement in order to increase performance [9]. Using our CHERI-VP, we study the performance impact of tag memory updates under representative embedded workloads and explore architectural optimizations aimed at reducing this overhead. In particular, we model an IMC-accelerated tag memory organization that reduces explicit load-modify-store sequences during tag updates.

Figure 2 shows an overview of our VP-based exploration flow for IMC-accelerated tagged memory optimization. Assuming IMC operation latency to be negligible compared to main memory access time, our early VP-based results indicate a measurable reduction of 6 % to 11 % in tag update overhead and overall execution time for CHERI-intensive workloads. During this exploration, we identified a lack of readily available benchmarks suitable for evaluating CHERI-enabled architectures in constrained embedded environments, as existing CHERI benchmarks often assume the presence of full operating systems such as CheriBSD. As a mitigation, we employed instruction stream generation with TestRIG [10] and applied additional CHERI-enabled post-processing to construct lightweight benchmark workloads that exercise capability manipulation and tag-memory behavior without requiring a full software stack.

Conclusion

This paper presented CHERI-VP, a framework for the early evaluation of CHERI in resource-constrained RISC-V systems. Through CHERI-VP, we explored verification workflows, obtained early performance insights, and identified practical challenges related to evaluating CHERI-specific mechanisms in embedded environments. Our experiences show that VPs are an effective tool for guiding early CHERI adoption decisions and architectural optimization efforts. Future work will extend these workflows with deeper CHERI-specific verification results and broader benchmark coverage.

References

[1] Carsten Bormann, Mehmet Ersue, and Ari Keranen. *Terminology for constrained-node networks*. Tech. rep. 2014.

[2] Microsoft Security Response Center (MSRC). *We need a safer systems programming language*. [Online]. 2019. URL: `https://www.microsoft.com/en-us/msrc/blog/2019/07/we-need-a-safer-systems-programming-language`.

[3] Yang Lu and Li Da Xu. "Internet of Things (IoT) cybersecurity research: A review of current research topics". In: *IEEE Internet of Things Journal* 6.2 (2018), pp. 2103–2115.

[4] Arsalan Mosenia and Niraj K Jha. "A comprehensive study of security of Internet-of-Things". In: *IEEE Transactions on emerging topics in computing* 5.4 (2016), pp. 586–602.

[5] Jonathan Woodruff et al. "The CHERI capability model: Revisiting RISC in an age of risk". In: *ACM SIGARCH Computer Architecture News* 42.3 (2014), pp. 457–468.

[6] Tom De Schutter. *Better Software. Faster!: Best Practices in Virtual Prototyping*. Synopsys Press, Mar. 2014.

[7] Manfred Schlägl, Andreas Hinterdorfer, and Daniel Große. "A RISC-V CHERI VP: Enabling System-Level Evaluation of the Capability-Based CHERI Architecture". In: *31st Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2026.

[8] MITRE. *CWE - Common Weakness Enumeration*. [Online]. URL: `https://cwe.mitre.org/`.

[9] Onur Mutlu et al. "Processing data where it makes sense: Enabling in-memory computation". In: *Microprocessors and Microsystems* 67 (2019), pp. 28–41.

[10] Alexandre Joannou et al. "Randomized Testing of RISC-V CPUs Using Direct Instruction Injection". In: *IEEE Design & Test* 41.1 (2024), pp. 40–49. DOI: `10.1109/MDAT.2023.3262741`.