Toward industrial-grade CHERI-enhanced RISC-V cores

Alexandre Joannou, Jonathan Woodruff, Peter Rugg,
Matthew Naylor, Franz A. Fuchs and Simon Moore

contact: firstname.lastname@cl.cam.ac.uk

Department of Computer Science and Technology, University of Cambridge

Abstract

CHERI-cap-lib is an RTL library implementing the core functionality necessary to extend CPU implementations with CHERI. We updated CHERI-cap-lib to the proposed 'Zcheri' CHERI RISC-V standard. We also enrich CHERI-cap-lib with formal property verification using SymbiYosys and develop a flow for formal equivalence checking between SystemVerilog implementations and the mature Bluespec SystemVerilog implementation.

Background

CHERI-cap-lib provides an API abstracting over the details of the CHERI capability in-memory format, as well as conversion functions to a specific in-register decompressed format which enables fast logic for use in processor pipelines.

The first CHERI-cap-lib implementation was derived from the original MIPS CHERI implementation. It was adapted and used in the Piccolo, Flute and Toooba RISC-V cores which were supporting the CHERI ISA development after the MIPS years. Having this library has enabled researchers to avoid repeating the error-prone and time consuming task of implementing every subtle aspect of the compressed CHERI capability format (CHERI Concentrate [1]) for each individual CPU extended for CHERI support, focusing the task to be an integration one rather than a complete redevelopment one, and crucially, benefitting from the verification effort that has already been poured in the library.

The CHERI RISC-V ratification effort

A CHERI RISC-V ISA extension ratification effort has been ongoing for the past few years. With the 'Zcheri' extension on the way, an updated version of CHERI-cap-lib is desirable.

The existing CHERI-cap-lib implements the University of Cambridge CHERI ISA V9 specification as opposed to the specification proposed at https://github.com/riscv/riscv-cheri. We developed CHERI-cap-lib to support the proposed 'Zcheri' specification to serve as a base for future CPU implementations aiming to support the new CHERI RISC-V extension.

Avoid ecosystem fragmentation

The CHERI-cap-lib code base is written in Bluespec System Verilog, a high level, functional HDL. The library can be used directly from within BSV code (which is the case for the Piccolo, Flute and Toooba cores). BSV can generate Verilog, making the library usable in a variety of other contexts as well. For example, if has been used used from System Verilog / Verilog cores (the CHERI-extended Ibex core), as well as in the Pebbles RISC-V processor framework (https://github.com/blarney-lang/pebbles).

The CVA6 open-source core has also seen an effort to augment it with CHERI support, but the lack of System Verilog / Verilog sources (available Verilog is generated from BSV sources) in CHERI-cap-lib motivated a re-implementation of the BSV-provided functionalities. We aim to audit and leverage this System Verilog work, and integrate it into CHERI-cap-lib for others to benefit from as they add CHERI support to other RISC-V cores.

Increase confidence in the implementation

The existing BSV CHERI-cap-lib has undergone lots of testing over many years. In particular, the TestRIG framework has been used extensively in the CHERI work. It leverages the Sail RISC-V (and CHERI RISC-V) model as a golden architectural model. Several CHERI capability properties have been formally verified within the Sail context. The TestRIG framework has leveraged this increased confidence and massively fuzzes implementations using the BSV CHERI-cap-lib against the sail model.

We recently contributed a flow using SymbiYosys to formally verify a variety of CHERI capabilities properties from the CHERI-cap-lib BSV sources directly. We are provide equivalence checking between the extensively tested BSV CHERI-cap-lib implementation and the System Verilog CHERI-cap-lib effort (we are able for example to verify whether the CVA6 provided CHERI functionality for CHERI V9 and those in BSV CHERI-cap-lib behave the same or not, and we are in the process of developing both a BSV and a SV CHERI-cap-lib version capturing the new proposed standard).

Conclusion

With the CHERI extension ratification effort coming to fruition, and the intricacies of the compressed CHERI capabilities being crucial to get right in a CHERI implementation, availability of quality open-source RTL is of first importance. We contribute to the effort of making such RTL available to the community and to the verification workflow to increase confidence in CHERI-cap-lib.

References

[1] Jonathan Woodruff et al. “Cheri Concentrate: Practical Compressed Capabilities”. In: *IEEE Transactions on Computers* 68.10 (2019), pp. 1455–1469.