

Verification of hardware and software usually proceeds separately, software analysis relying on the correctness of processors executing instructions. This assumption is valid as long as the software runs on standard CPUs that have been extensively validated and are in wide use. However, for processors exploiting custom instruction set extensions to meet performance and energy constraints the validation might be less extensive, challenging the correctness assumption.

We present an approach for integrating software analyses with hardware verification, specifically targeting custom instruction set extensions. We propose three different techniques for deriving the properties to be proven for the hardware implementation of a custom instruction in order to support software analyses. The techniques are designed to explore the trade-off between generality and efficiency and span from proving functional equivalence over checking the rules of a particular analysis domain to verifying actual pre and post conditions resulting from program analysis. We demonstrate and compare the three techniques on example programs with custom instructions, using state-of-the-art software and hardware verification techniques.