

A Mathematical Approach to RTL Verification (Invited Talk)

David M. Russinoff

Advanced Micro Devices, Inc.
david.russinoff@amd.com

The formal hardware verification effort at Advanced Micro Devices, Inc. has emphasized theorem proving using ACL2, and has focused on the elementary floating-point operations. Floating-point modules, along with the rest of our microprocessor designs, are specified at the register-transfer level in a small synthesizable subset of Verilog. This language is simple enough to admit a clear semantic definition, providing a basis for formal analysis and verification. Thus, we have developed a scheme for automatically translating RTL code into the ACL2 logic, thereby reducing the potential for error in the development of formal hardware models.

Formal statements of correctness (IEEE compliance) of arithmetic operations are encoded in the same language and translated into ACL2 along with the RTL. Their proofs are developed interactively and mechanically checked with the ACL2 prover.

Much of the effort involved in this project has been in the development and formalization of a general theory of floating-point arithmetic and its bit-level implementation, resulting in an ACL2 library of lemmas pertaining to bit vectors, logical operations, floating-point representations, and rounding. The library is publicly available as a part of the standard ACL2 release.

In this talk, I will describe my experience over the past decade in the development and application of this methodology. I will describe lessons learned through the process, especially regarding the relevance of the established principles and methodologies of both software verification and traditional mathematics to the hardware problem. Finally, I will discuss prospects for extending these methods to functional areas beyond the floating-point unit and the ultimate objective of a fully verified microprocessor design.