

Formal Verification for Practical Design Flows

Keywords

Formal Verification, Digital Hardware, Correctness proofs, System Evolution&Optimization, Eliminate human design errors

Goal

We consider the problem to ensure the correct functionality of digital systems. For this purpose we plan to develop formal verification techniques that are practical to use by non-experts and that can be integrated into the design and evolution phases of digital systems.

Motivation&Problem

With the presence of digital systems in almost any technical system, the correct functionality of digital systems is of high importance to the general public. The common use of digital systems even in life critical environments further emphasizes this importance of the correctness of digital systems. Current digital systems are too complex to guarantee their correct behavior just by testing. Formal verification has therefore been used to mathematically prove the correctness of digital systems -- showing that a particular implementation is in full accordance with its specification. In general, formal verification is performed by semi-automatic procedures. The non-automatic parts need to be handcrafted by an expert particular to the system considered. Conventional hardware designers are generally not responsible for and able to deal with formal verification. The design and verification is considered by two different groups and already the communication and translation of the design and specification from one group (and description language) to the other can insert or hide inconsistencies. To avoid such problems and to reduce overhead, it is obvious that the verification of digital systems should be more closely integrated into the design and development phases of digital systems.

Approach

We propose to develop formal verification techniques that can be integrated into practical design flows. We are proposing the following approaches: (1) Digital systems are never designed just by trial and error. The hardware designer has to have a good intuitive explanation for the correctness of his design. If the argumentation of the designer is translated into a formal description, the reasoning based on the intuitive explanation can be cleaned from human error. We propose to develop simplified interfaces for the translation of correctness arguments to guide formal verification procedures. (2) The involvement of an expert in the formal verification of a particular digital system can be applied similarly to similar systems. We are evaluating the limitations and opportunities of fully automated verification for a parameterized family of digital systems. Verified components are made available through a library and interfaces are built for the composition and customization of verified components. (3) Digital systems are rarely developed from scratch, but they evolve in numerous iterations. Most changes between iterations steps can be considered as the application of common optimization techniques. We target the evolution of correctness proofs to adapt to the evolution of the digital systems. This avoids having to deal with formal verification from scratch in each design iteration.