Abstract

Large digital integrated circuits typically consist of multiple functional units integrated using glue logic. The main functional units in the design are typically extensively verified using both formal and simulation techniques. On the other hand, the glue logic, which accounts for a large fraction of the integrated circuit (40% or above), is often developed from an intuitive understanding of the architecture, typically not documented properly and not formally verified in industrial practice. Consequently many of the late bugs are uncovered in this part of the design, and are associated with incorrect interpretation or implementation of the micro-architectural specification.

The glue logic is often modified locally during the design cycle for fixing these late bugs. However, these local changes may indirectly affect a much larger portion of the glue logic, and it is non-trivial to determine the exact boundary of the cone-of-influence of that change. The task of using formal methods to verify whether the functionality of the modified glue logic in its entirety remains unchanged even after application of these local design changes, is a very hard problem in practice considering the enormity of the glue logic and the nature of the cone of variables influencing the local design change. Model checking or sequential equivalence checking on the entire glue logic as a whole, does not scale. In this work, we propose effective methodologies for verifying such local design changes, without attempting to apply formal methods on the entire glue logic. Specifically we solve the following problems:

1. Trace Assisted Formal Methods for the Verification of Bug Fixes: Bug traces reproduced in simulation serve as the basis for patching the RTL code which is essentially a local design change in order to fix the bug. This work proposes formal methods inspired from software debugging for analyzing the control trace obtained from the given bug trace on the fixed design, and for verifying the robustness of the bug fix with respect to that control trace and providing formal guarantees with respect to the specific bug scenario.

2. Formal Methods for Ranking Counterexamples: Verifying local design changes on the total glue logic involves cutting out a cone-of-influence of the change. Verifying the property on that cone-of-influence in isolation typically throws up a large number of counterexamples, many of which are spurious because the scenarios they depict are not possible in the entire logic. In this work, we introduce the notion of ranking the counterexamples and assigning confidence values to them so that only the most likely counterexamples are presented to the designer.

3. Reusing Component Invariants in Equivalence Checking: This work explores the utility of reusing proven component invariants in the backward reachability-based sequential equivalence checking paradigm of formal verification, for verifying that the modified glue logic is equivalent to the previous version.

The thesis also presents the empirical evaluation of the proposed techniques using standard benchmark circuits and the commercial OpenSPARC T1 processor design.

It is anticipated that the research presented in this thesis will motivate the industry to adopt some of these methods into industrial verification practices.

Keywords: Verification, Sequential Equivalence Checking, Simulation, Formal Verification, Model Checking, Properties, Traces, Counterexamples