DRILLS: Debugging RTL Intelligently with Localization from Long Simulation

Donggyu Kim, Vighnesh Iyer

The goal of this project is to develop a productive debugging tool for hardware designs. Specifically, we will develop a methodology to localize the origin of bugs in hardware designs given a runtime assertion failure and an error trace leading up to the failure.

As hardware complexity increases to meet performance targets and implement required functionality, verification becomes much more challenging. A recent study shows that verification dominates time-to-market and it is getting worse over time[1]. Therefore, it is critical to invent effective hardware verification tools to alleviate time and manual effort to trace the root cause of a design bug.

Simulation-based verification has been the most effective technique for system-level verification. To check whether or not the whole system works for real-world workloads, the hardware design is emulated using an FPGA for trillions of cycles. FPGA-based simulation is much faster than software simulation but suffers from a lack of DUT visibility. DESSERT[2] demonstrates a technique to catch errors and obtain complete waveform traces from FPGA-based simulation with assertion synthesis and commit log comparisons. However, only violations of high-level properties are caught, which need to be manually traced back to the originating bug in the source RTL.

There has been plenty of prior work on SAT-based bug localization[3][4][5]. The general idea is to instrument the DUT with muxes for suspect lines of RTL, transform the instrumented hardware design into a CNF formula, and let a SAT solver pick out lines of RTL which could produce a bug. However, this approach does not scale with complex hardware designs and long error traces.

In this project, we propose a novel methodology to effectively localize bugs from error traces. Our proposal is to localize bugs using fine-grained specifications which are mined from error-free traces. We will employ the mined specifications on error traces that were produced by catching violations of high-level properties, in the hope that the fine specs will catch design errors before the high-level assertions are triggred and point to specific lines of potentially buggy RTL.

We will collect traces from small tests as well as realistic workloads from DESSERT. Using these traces, we will employ template-based spec-mining suggested by Li et. al[6]. Since this approach is computationally efficient, we can derive many simple, but fine-grained assertions from long traces. Note that we do not have to merge these simple assertions into more complex properties since they are only used for bug localization. We will explore adding more specification templates that can better localize bugs.

The timeline for this project is as follows:

1. **March**: Vighnesh will develop a tool to convert VCD dumps into module-level delta traces. Donggyu will figure out what templates should be introduced for effective bug localization. We believe specifications that take into account the module's cycle-level behavior in terms of absolute cycles will be helpful.

2. **April**: We will implement a specification-mining tool using module-level delta traces, and apply this tool to simple designs. We will also collect error-free and error traces for complex designs using DESSERT.

3. **May**: We will present preliminary results on complex designs in the class. We will continue this work following this semester and plan to publish this work.

References

[1] H. D. Foster, "Trends in Functional Verification: A 2014 Industry Study," DOI: `10.1145/2745404.2751548Technical`. [Online]. Available: `http://dx.doi.org/10.1145/2745404.2751548Technical`.

[2] D. Kim, C. Celio, S. Karandikar, D. Biancolin, J. Bachrach, and K. Asanovic, "DESSERT: Debugging RTL effectively with state snapshotting for error replays across trillions of cycles," in *Proceedings - 2018 International Conference on Field-Programmable Logic and Applications, FPL 2018*, 2018, ISBN: 9781538685174. DOI: `10.1109/FPL.2018.00021`.

[3] A. Veneris, "Fault diagnosis and logic debugging using Boolean satisfiability," in *Proceedings - International Workshop on Microprocessor Test and Verification*, 2003, ISBN: 0769520456. DOI: `10.1109/MTV.2003.1250264`.

[4] K. H. Chang, I. Wagner, V. Bertacco, and I. L. Markov, "Automatic error diagnosis and correction for RTL designs," in *Proceedings - IEEE International High-Level Design Validation and Test Workshop, HLDVT*, 2007, ISBN: 1424414806. DOI: `10.1109/HLDVT.2007.4392789`.

[5] S. Mirzaeian, F. Zheng, and K.-t. Tim Cheng, "RTL Error Diagnosis Using a Word-Level SAT-Solver," Tech. Rep.

[6] W. Li, A. Forin, and S. A. Seshia, "Scalable specification mining for verification and diagnosis," 2010. DOI: `10.1145/1837274.1837466`.