A New Trends in RTL Verification: Bug Localization, Scan-Chain-Based Methodology, GA-Based Test Generation

Khaled Salah
Mentor Graphics

RTL BUG LOCALIZATION

The proposed methodology for automated bug localization

THE PROPOSED RTL-LEVEL SCAN-CHAIN METHODOLOGY

Proposed Emulation Flow (online flow), synthesizable testbench methodology, scan-chain methodology, a) detailed, (b) simplified.

(a) Normal design example, (b) proposed scan-chain methodology for the design example in (a).

GA-BASED PROPOSED METHODOLOGY

The proposed GA methodology to speedup coverage closure. Using genetic algorithms, there is no test redundancy

The GA performance

Conclusions

- Bug localization is a process of identifying the specific locations or regions of source code that is buggy and needs to be modified to repair the defect.
- Bug localization can significantly reduce human effort and design cost.
- In this paper, a novel automated coverage-based functional bug localization method for complex HDL designs is proposed which significantly reduces debugging time.
- The proposed bug localization methodology takes information from regression suite as an input and produces a ranked list of suspicious part of code.
- Our methodology is a promising solution to reduce required time to localize bugs significantly.
- Moreover, an online RTL-level scan-chain methodology is proposed to reduce debugging time and effort for emulation.
- Run-time modifications of the values of any of the internal signals of the DUT during execution can be easily performed through the proposed online scan-chain methodology.
- A utility tool was developed to help ease this process.
- Our experiment shows that, the area overhead is neglected compared to the gained performance benefits. But, IP design requires more compilation time.
- The main challenge in using constraint random testing (CRT) is that manual analysis for the coverage report is needed to find the untested scenarios and modify the test cases to achieve 100% coverage.
- We need to replace the manual effort by an automatic method or a tool that will be able to extract the coverage report, identify the untested scenarios, add new constraints, and iterate this process until 100% coverage is attained.
- In this paper, the implementation of this automatic feedback loop is presented.
- The automatic feedback loop is based on artificial intelligence technique called genetic algorithm (GA).
- This technique accelerates coverage-driven functional verification and achieves coverage closure rapidly by covering uncovered scenarios in the coverage report (coverage holes).