

IC/CAD Contest

Problem 1: HDL Code Coverage

Source: SpringSoft Inc.

February 15, 2000

1. Introduction

As VLSI design complexity grows, correctness of circuit functionality becomes more difficult to verify. For designers, they can easily develop test patterns, however, they cannot guarantee to run through all parts of design. CAD tools can help designers solve this problem by checking which parts to be run through and which parts not to be traversed by given test patterns. Moreover, after building the relationship of signal flow (gate level) or design flow (behavior level) and deducing process of code coverage, extracting extra minimum test patterns to cover the parts of not being traversed could be expected.

2. Problem Description

The basic requirement of this problem is to develop a tool that measures the **HDL code coverage** during a simulation. Your software can be integrated into some existing simulators or self-contained by itself.

3. Input

- (1) HDL language: simple subset of any known HDL language.
- (2) Any design using HDL language in 1. Design could be flattened or hierarchical.

4. Basic Output

< Basic report >

- (1) Show the hierarchy of the design under test.
- (2a) Differentiate the traversed modules and the modules not to be traversed.
- (2b) Show the code coverage using modules as the basic unit.
- (3a) Differentiate the traversed codes and the codes not to be traversed when designers selects specific module to view detail.
- (3b) Show the percentage of traversed codes for each traversed modules using statement coverage metric.

< Advanced report >

- (1) Show code coverage on different coverage metrics. See papers [1, 2, 3].

5. Advanced Features / Functions

The above tool only can announce the generated test pattern will run through all modules, but has no idea about the fault coverage of this test set. Actually, tool can build fault-detecting

information during the process of deducing code coverage, and then deduce the detected/undetected fault lists and derive extra test patterns for undetected faults.

< Basic report >

- (1) Show the detected faults list and the undetected faults list according to existing test patterns based on the statement coverage metric.
- (2) Show the derived test patterns for those undetected faults.

<Advanced report>

- (1) Show the detected faults list and undetected faults list according to existing test patterns based on other coverage metrics.
- (2) Show the derived test patterns for those undetected faults.

6. Language / Platform

- Language: C or C++.
- Platform: SUN OS / Solaris or PC DOS / Windows.

7. Evaluation

- Correctness, advanced features / functions, user interface, etc.

8. Technical Support

Solving this problem may require using a HDL parser. To request for the parser, please contact Springsoft.

- Ting Chen(陳調鈺副總經理): Email: ting@springsoft.com.tw ; Tel: (03) 5794567 ext. 312.

9. Questions

Please report any question regarding this problem to cad@cis.nctu.edu.tw with the email subject "CAD Contest: Problem 1." Your question(s) will be answered in two weeks, and the Q&A's will be posted at the contest web site.

References

- [1] Jing-Yang Jou and Chien-Nan Jimmy Liu, "Coverage Analysis Techniques for HDL Design Validation", the 6th Asia Pacific Conference on cHip Design Languages (APCHDL'99), October 1999. (Invited Paper)
- [2] Chien-Nan Jimmy Liu, Chen-Yi Chang, Jing-Yang Jou, Ming-Chih Lai and Hsing-Ming Juan, "A Novel Approach for Functional Coverage Measurement in HDL", International Symposium on Circuit And Systems (ISCAS 2000), May 2000.
- [3] D. Drako and P. Cohen, "HDL Verification Coverage", Integrated Systems Design Magazine, June 1998.
(<http://www.isdmag.com/Editorial/1998/CodeCoverage9806.html>)