Problem 6: Improving Test Pattern Quality Using Multiple Detection

2005 ICCAD/SIGDA CADathlon

1. Introduction

Conventional manufacturing test techniques generate one test pattern per fault, and attempt to achieve as high a coverage as possible. Multiple−detect (or n−detect) test generation is an emerging manufacturing test strategy where each fault is detected by n different test vectors in the test set. This has been shown to greatly improve the quality of stuck−at test sets, and increases the ability to detect unmodeled defects. To solve this problem, you have to use your creativity to modify a basic ATPG engine to generate 5−detect test sets. The performance of your solution will be evaluated using the bridging coverage estimate (BCE) metric as well as test set compactness.

2. Problem Statement

2.1. Brief Description

Given a single stuck−at fault test pattern generator and a fault simulator, convert it to a n−detect test pattern generator with n = 5. The quality of your test patterns will be measured using (i) the BCE metric described in the reference paper and (ii) the size (number of vectors) of the test set that you generate. The theoretical BCE limit for a 5−detect test pattern set is 96.875% − once this limit is reached by your solution, the next step would be to reduce the number of patterns required to reach this BCE value. In solving this problem, remember that higher BCEs are good. However, test pattern sets are also evaluated on the basis of the total number of patterns in the real world. So, if there is more than one team that reaches the BCE limit for a test case, the total number of test patterns that were generated to reach this limit will be considered. Partial credit to all other qualifying solutions (ones that improve BCE) will be awarded on a normalized basis with respect to the best solution at the discretion of the judges.

2.2. Input/Output Specification

You may assume that the supplied test cases are well−formed and can be read without errors with the provided parser.

2.2.1. Input

The main executable is called testgen, and it requires two input files. The circuit description file (**ckt.i**) is the first file. The second file is the fault list file (**ckt.dfc**) − it has the list of all the deductible faults in the circuit. Both these files will be supplied for each test case and the routine to read them in are already provided. No modifications are required. You will see some statistics reported on the screen when you run testgen. For example, on test case 1, testgen produces

```
Total vectors: 72
Total number of faults: 379
Undetectable faults: 0
Aborted faults: 0
```

Fault coverage: 100.00% (379)

This clearly tells you that the generated test set detects all 379 faults (100% fault coverage) with 72 test vectors.

2.2.2. Output

The output of testgen is the (at least) 1−detect test pattern set. It consists of one test pattern on each line. This test pattern file is provided as an input to the BCE routine.

2.2.3. BCE

The test pattern set generated by testgen is provided to the BCE executable for evaluation using the 5−detect metric. In addition to the .i and .dfc files, the BCE routine accepts the test pattern set and produces the following output:

```
Total vectors: 72
Total faults detected: 379 (100.00%)
BCE = 84.33
```

Your task is to modify testgen to maximize the BCE value with as small a test set as possible, while still maintaining 100% fault coverage.

We will judge your solution as follows: If your solution does not have 100% fault coverage it is discarded and you do not get any credit; If you have the highest BCE value of all teams you get maximum credit; In case other teams have the same high BCE value, then ties are broken by looking at the number of test patterns; the lower the number the better.

3. Additional information

Familiarize yourself with the data structures and convenience functions in file "**define.h**". The file "**sim.c**" has functions related to fault simulation. "**ckt.c**" has input/output functions. "**podem.c**" has the test pattern generation files.

4. References

- "Impact of Multiple−detect Test Patterns on Product Quality", B. Benware, C. Schuermyer, S. Ranganathan, R. Madge, P. Krishnamurthy, N. Tamarapalli, K.−H. Tsai, and Janusz Rajski, *Proc. International Test Conference (ITC),* pp. 1031−1040, 2003.