EEIE30069: VLSI Testing

Assignment #6 (Due: Nov. 25 , 2024 23:59:99)

Last update: Nov. 11, 2024

Reading:

- Chapter 7 Combinational Circuit Test Generation (Bushnell and Agrawal), or Section 4.1-4.5 (Wang, Wu and Wen)
- b17.bench (sequential circuit break into combinational circuit)
- Please put all results into a report and try to discuss them
- Please compress the folder containing all source code files and a report into a file named `ASS6_<your-student-ID>` and upload it to new E3
- The report should cover
  - The algorithm or idea of your code
  - Several case results
  - Discuss your results
  - How to compile your code
  - Other information (optional)

Homework Description:

- (200 pts) **Combinational test generation**
  a. (20 pts) Generate test vectors for b17.bench with the PODEM program. Set backtrack limit to 1, 10, 100, 1000, or more and compare number of patterns, fault coverages, CPU run times, and actual backtrack numbers generated by running the program. (target circuit: b17.bench)
  b. (30 pts) (Verifying ATPG results) Generate test vectors for b17.bench benchmark circuit with collapsed fault list (checkpoint fault list) and total fault list. Then run the fault simulator on the total fault list with the two sets of generated patterns. Is there any difference? If so, could you explain what've happened? (target circuit: b17.bench)
  c. (50 pts) (PODEM Implementation) First, trace the ATPG part of the PODEM program, and try to understand the implementation of the test generation procedure. Then use c17.bench circuit to demonstrate the PODEM procedures (print to stdout by inserting `printf` or `cout`). Selected fault list is listed below and you have to exercise most procedures used in the ATPG. It is recommended to use a table for the program outputs to show each step (fault activation, fault propagation, backtrace, forward simulation, backtrack, etc.), and a decision tree will be helpful for viewing the process. (target circuit: c17.bench) fault_list
  d. (50 pts) (ATPG with random patterns) Use a random pattern generator as the first stage of a ATPG system. The random pattern generator stops at either a fixed number

of patterns (1000 patterns) or a saturated fault coverage (90%). After the random generator, PODEM will be used to target the remaining undetected faults. Please compare the results with original PODEM. Note that the random pattern stage and ATPG stage have to be integrated such that you can measure the corresponding CPU times. (target circuit: b17.bench, s35932_com.bench, s38417_com.bench, s38584_com.bench)

e. (50 pts) (Test generation for bridging faults) Please modify PODEM program to deal with the bridging fault list produced in homework #4-1.-b. Verify your patterns with the fault simulator implemented in assignment #5.

Grading:

- **Correctness 90%**
- **Report 10%**

Attention:

- Please just output in the original output format of PODEM ATPG.
- Command for d.:

  ./atpg -random_pattern -output [output_pattern] [circuit_name]

- Command for e.:

  ./atpg -bridging_atpg -output [output_pattern] [circuit_name]